ADAPTIVE CONTROL TECHNIQUES FOR TRANSITION-TO-HOVER FLIGHT OF FIXED-WING UAVS

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Aerospace Engineering

by

Brian Decimo Marchini

December 2013

COMMITTEE MEMBERSHIP

TITLE:	Adaptive Control Techniques for Transition-to-Hover Flight of Fixed-Wing UAVs
AUTHOR:	Brian Decimo Marchini
DATE SUBMITTED:	December 2013
COMMITTEE CHAIR:	Eric Mehiel, Ph.D., Associate Professor, Aerospace Engineering Department
COMMITTEE MEMBER:	Robert McDonald, Ph.D., Associate Professor, Aerospace Engineering Department
COMMITTEE MEMBER:	John Ridgely, Ph.D., Professor, Mechanical Engineering Department
COMMITTEE MEMBER:	Charles Birdsong, Ph.D., Professor, Mechanical Engineering Department

ABSTRACT

Adaptive Control Techniques for Transition-to-Hover Flight of Fixed-Wing UAVs

Brian Decimo Marchini

Fixed-wing unmanned aerial vehicles (UAVs) with the ability to hover combine the speed and endurance of traditional fixed-wing flight with the stable hovering and vertical takeoff and landing (VTOL) capabilities of helicopters and quadrotors. This combination of abilities can provide strategic advantages for UAV operators, especially when operating in urban environments where the airspace may be crowded with obstacles. Traditionally, fixed-wing UAVs with hovering capabilities had to be custom designed for specific payloads and missions, often requiring custom autopilots and unconventional airframe configurations. With recent government spending cuts, UAV operators like the military and law enforcement agencies have been urging UAV developers to make their aircraft cheaper, more versatile, and easier to repair. This thesis discusses the use of the commercially available ArduPilot open source autopilot, to autonomously transition a fixed-wing UAV to and from hover flight. Software modifications were made to the ArduPilot firmware to add hover flight modes using both Proportional, Integral, Derivative (PID) Control and Model Reference Adaptive Control (MRAC) with the goal of making the controllers robust enough so that anyone in the ArduPilot community could use their own ArduPilot board and their own fixed-wing airframe (as long as it has enough power to maintain stable hover) to achieve autonomous hover after some simple gain tuning. Three new hover flight modes were developed and tested first in simulation and then in flight using an E-Flight Carbon Z Yak 54 RC aircraft model, which was equipped with an ArduPilot 2.5 autopilot board. Results from both the simulations and flight test experiments where the airplane transitions both to and from autonomous hover flight are presented.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Traditionally unmanned aerial vehicles (UAVs) have been split into two categories: fixed-wing aircraft that perform higher altitude, longer endurance missions, and rotary aircraft (including both traditional helicopters and multi-copters) that perform shorter missions where low altitude, precision flying is required. This is mainly due to the fact that fixed-wing aircraft typically have higher top speeds, longer endurance, and higher altitude limits than rotary wing aircraft but also require more airspace to maneuver and large, open spaces to land and takeoff. On the other hand, most rotary aircraft have vertical takeoff and landing (VTOL) capabilities, as well as the ability to hover in place, which makes operating in crowded urban environments much easier but hurts the vehicle's overall endurance and speed. UAV manufactures have attempted to work around these limitations in different ways. For example, Aerovi-

Figure 1.1: Aerovironment's Raven UAV

ronment designed their backpack transportable surveillance UAV named Raven [3], shown in Figure 1.1, to be launched using a hand toss method and land using a deep stall maneuver to help reduce the ground area needed to operate the UAV, while still maintaining the speed and range of a fixed-wing aircraft. Bell Helicopters takes the

Figure 1.2: Bell Helicopter's Eagle Eye UAV

opposite approach with their custom designed Eagle Eye tilt-rotor UAV [27], shown in Figure 1.2, that flies like a helicopter for takeoff, landing, and precision flying but converts to a fixed wing aircraft for long distance portions of the mission. Both of these vehicles, however, require specialized hardware and custom control systems to handle the unique maneuvers they are asked to perform.

Fixed-wing UAVs with the ability to hover combine the speed and endurance of traditional fixed-wing flight with the stable hovering and vertical takeoff and landing (VTOL) capabilities of helicopters and multi-copters. This combination of abilities can provide strategic advantages for UAV operators, especially when operating in urban environments. The first inspiration for fixed-wing aircraft with hovering capabilities came out of fear of a Soviet invasion of Western Europe in the days shortly after World War II. The American military feared that if the Soviets did invade Western Europe the first thing they would do is take over the Allied forces runways and airstrips to knock out the Allied air support. Having aircraft that could take off and land vertically would give the Allies the ability to maintain air superiority even if they were unable to keep control of their airfields. In 1954, the Convair XFY-1 "Pogo" [25], shown in Figure 1.3, became the first tail-sitter prototype to successfully

Figure 1.3: Convair XFY-1 "Pogo" Tail-sitter Prototype.

take off vertically, transition to level flight, and then land vertically all in the same flight. Since then, technology advancements in aircraft structures and propulsion systems have allowed tail-sitters to be designed more like their conventional fixed-wing counterparts. Aerovironment's recent SkyTote UAV prototype [2], shown in Figure 1.4, uses a gas powered engine to turn two coaxial contra-rotating propellers, but has a main wing and vertical/horizontal stabilizers to help improve cruise efficiency compared to the XFY-1's delta wing configuration.

1.1 Literature Survey

The most recent advancements in the field of fixed-wing hovering have come from the radio-controlled (RC) aircraft world. New lithium-polymer (LiPo) batteries and high power electric motors, as well as new structural foams and composites, have

Figure 1.4: Aerovironment's SkyTote UAV

made it much easier to design fixed-wing aircraft with thrust-to-weight ratios greater than one. As long as an aircraft has a thrust-to-weight ratio greater than one, and large enough control surfaces to maneuver based on propeller down-wash alone, any fixed wing aircraft theoretically has the potential to maintain stable hover.

Various academic institutions have started using these new high powered stock RC airframes to take different approaches at expanding the capabilities and reliability of hovering fixed-wing UAVs. Frank, McGrew, Valenti, Levine, and Jonathan of MIT [14] do all of their research indoors using a small foam aircraft (less than 3 foot wingspan) and a motion capture system, similar to what is used in the movie industry. The motion capture system allows them to do all of their computing and control off board the airplane while having complete situational awareness of the aircraft at all times. While this limits their work to the confines of the motion capture room, the highly accurate motion capture system minimizes sensor noise and allows for unbiased comparison when testing different transition-to-hover methods and control algorithms. Johnson, Turbe, Wu, Kannan and Neidhoefer of Georgia Tech [16], on the other hand, use a large (8 foot wingspan) gas powered RC airplane equipped with a custom built autopilot system to do all of their testing outdoors and all of the autopilot computing on board the aircraft. Instead of using highly accurate sensors,

they use a dynamic inversion algorithm to add robustness to the controller and help account for the wind disturbances that are unavoidable in outdoor flight.

Other research groups have approached the fixed-wing hover problem in more specific ways. Yeo, Henderson, and Atkins of the University of Michigan [31] added an aerodynamic data system to the standard set of accelerometers and gyroscopes in their autopilot to provide the controllers information on whether flow over the wings has stalled in order to control gain scheduling while the aircraft is in transitional flight states. Similarly, Matsumoto, Kita, Suzuki, Oosedo, Go, Hoshino, Konno, and Uchiyama of Tohoko University [20] attempted to make their hovering aircraft more resistant to disturbances by redefining the way the airplane's attitude is measured relative to vertical hover.

Figure 1.5: Stanford's Perching UAV

Some other works have looked at expanding the applications of fixed-wing hover flight beyond that of just vertical takeoff and landing. Since many small UAVs are used for surveillance, many of the works mentioned above talk about the potential for hovering flight to provide a stationary airborne surveillance platform so that the same aircraft could survey broad areas while in level flight and provide pinpoint

surveillance of small areas, such as an intersection in a crowded city, while in hover flight. Researchers at Stanford [8] have taken this concept a step further by using hover and deep stall maneuvers to "perch" a fixed-wing UAV on a wall, or the side of a building (shown in Figure 1.5), by using specially designed claws that are attached to the belly of the aircraft. The concept is based on the fact that a perched UAV could provide the same pinpoint surveillance as a hovering UAV, but do it for much longer periods of time because the motor can be shut off while the UAV is hanging on the side of a building. Similarly, researchers at MIT [23] have used the same principle to hook a UAV onto a power line with the intent of using the power line to recharge the aircraft's flight batteries. Both of these techniques have the potential to greatly increase the usefulness and and endurance of small fixed-wing UAVs in real world applications.

1.2 Problem Definition

This thesis approaches the problem in a different way by attempting to make autonomous fixed-wing hovering more accessible by using a well documented, low cost, and widely used open source autopilot called the ArduPilot-Mega (APM) [12]. While the works mentioned above all do a good job of standardizing their hardware within their own institutions and research groups, the custom autopilots and airframes they use make it difficult to replicate their results and make use of them in real world UAV applications. By adding fixed-wing hover capabilities to an open source autopilot with a large community of developers and users, the potential for use in real world applications (even if it is only for fun) and further development is greatly increased.

In addition to making hovering capabilities more accessible to autopilot users, this thesis also attempts to make the capability more user friendly. Many of the previous attempts at autonomous fixed-wing hovering, including [7], [15], [24], and [31], use

classical control techniques, like pole-placement, to achieve their desired transition and hover performance. While these methods do work well and have been used for many years, they require extensive knowledge of the plant and classical control theory in order to get adequate performance, which would make using the autopilot difficult for hobbyists and civilian users.

This thesis attempts to use two different types of controllers to achieve adequate transitional and hovering flight performance while maintaining relative ease of use. The first is a set of Proportional, Integral, Derivative (PID) controllers that are tuned using simple trial and error techniques to control the orientation of the aircraft. These PID controllers are the standard method that the APM uses to control any aircraft in level flight, so they are used as the baseline controllers for hover performance. The second controller is called a Model Reference Adaptive Controller (MRAC) which works by mathematically modifying the control gains in flight to help compensate for any control deficiencies in the initial user defined set of control gains, or any unexpected changes to the flight conditions. These could include small changes in wind direction or magnitude, changes in air temperature or altitude, or possibly even the use of a different airframe compared to the one that the initial gain tuning was performed in. The MRAC method will hopefully make the hover controller more robust in the presence of disturbances, and also increase the precision of the transition-to-hover maneuver.

This thesis will also attempt to demonstrate the ability of the autopilot to transition back to level flight, from hover orientation, using one of the standard ArduPilot flight modes to prove that the firmware modifications necessary to achieve hover flight do not ruin the autopilot's ability to perform as a normal fixed-wing aircraft.

1.3 Overview of Thesis

First, details on the ArduPilot-Mega autopilot system and the E-flite Carbon-Z Yak 54 airframe used for flight testing are presented in Chapter 2. Chapter 3 discusses the relationship between Euler angles and quaternions. Here, more information about how the standard APM firmware stabilizes the aircraft is presented as well as some of the changed that had to be made in order for the autopilot to be capable of controlling a stable hover. Next, detailed descriptions for both the PID controllers and MRAC are presented in Chapter 4, including details about the modified APM control logic and how it implements the two controllers. Chapter 5 discusses the methods used to simulate transition-to-hover flight using Matlab and Simulink. Equations and methods used for aerodynamic force and moment calculations for the wings, stabilizers and fuselage are presented as well as those for a first order propulsion system model. Chapters 6 and 7 then present simulation and flight test results, respectively, for both the PID and MRAC methods for transition-to-hover, as well as the results for the transitions back to level flight. Chapter 6 also includes details on how simulation conditions were chosen and how disturbances were introduced to help mimic the actual flights and help develop controller robustness. Finally, this thesis will be ended with a chapter covering concluding remarks as well as a section on future ares of work that could be pursued in relation to the field of fixed-wing hovering UAVs and the MRAC theory.

2 AUTOPILOT AND AIRFRAME

Detailed descriptions of both the ArduPilot-Mega autopilot system and the E-Flite Carbon-Z Yak 54 RC aircraft model used in flight testing are presented here. Product specifications for both the autopilot and aircraft are shared, along with reasoning as to why both were chosen for use in this thesis.

2.1 ArduPilot-Mega 2.5 Autopilot System

The ArduPilot-Mega (APM) [12], developed by DIY Drones and manufactured by 3D Robotics, Inc., is a low cost, easy to use autopilot system designed to work with fixed-wing and rotary aircraft, as well as both land and water based ground vehicles. This level of versatility is accomplished by having a single autopilot board that is capable of running multiple types of firmware that are each specialized for a specific type of vehicle. These different firmware versions include ArduPlane, ArduCopter, and ArduRover which are designed for fixed-wing aircraft, rotary aircraft, and ground vehicles respectively. The firmware itself is written in a modified version of C++ which uses both standard Arduino libraries [6] as well as custom libraries specific to the ArduPilot project. Development of the APM and its firmware is mainly done by the DIY Drones Development Team, but since the firmware is open source and the APM boards are designed to work with standard radio controlled (RC) vehicle equipment (servos, receivers, transmitters, etc.), hobbyists around the world have the ability to make their own improvements and contribute to the ArduPilot community. The DIY Drones website [11] allows the DIY Drones Development Team to interact with the rest of the community through forums, manuals and blog postings to answer questions, fix bugs, and continually develop the ArduPlane software.

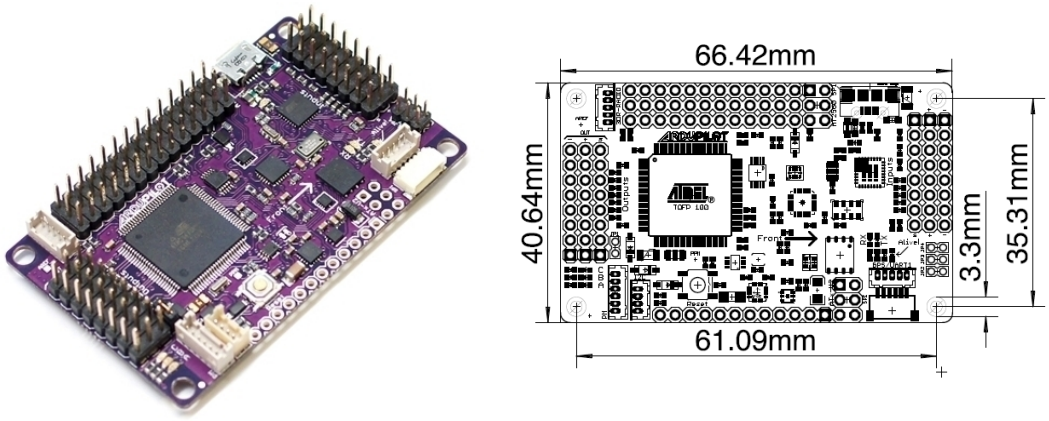


Figure 2.1: ArduPilot-Mega 2.5 Board

The APM has gone through several hardware iterations in addition to the constantly evolving ArduPlane firmware. In order to eliminate the risk of hardware changes and outside firmware changes influencing the results of the work done for this thesis, APM version 2.5 and ArduPlane version 2.66 were used as the standard hardware and firmware combination throughout this project.

The APM 2.5 autopilot board, shown in Figure 2.1, is equipped with an Atmel ATMEGA2560 microcontroller processor. Its sensor package includes an InvenSense MPU-6000 Six-Axis (Gyro + Accelerometer) MEMS Motion Tracking Device, a Honeywell HMC5883L-TR 3-Axis Digital Compass, and a Measurement Specialties MS5611 MEAS High Resolution Altimeter. The standard APM 2.5 system, shown in Figure 2.2 is also shipped with an external Mediatek MT3329 GPS module and an APM Power Module that delivers a clean 5 V power supply to the board from the main flight battery. The APM 2.5 can also be equipped with an optional pitot-static airspeed sensor but this add-on was not used for any of the controller development or flight tests in this thesis. The optional pitot-static sensor is not necessary for the APM to perform any of its standard level flight functions, so the modifications made for hover flight were designed not to need it as well. The APM

2.5 was equipped, however, with the optional 915 MHz 3DR Radio Telemetry Kit so that detailed information about the aircraft and the autopilot could be monitored in flight.

Figure 2.2: APM 2.5 autopilot system includes the APM 2.5 microcontroller board, the APM Power Module, and an external GPS module.

The Mission Planner software, shown in Figure 2.3, is specifically designed to work with any of the APM versions, even if custom firmware is uploaded to the board. Without a telemetry link, the APM board can be connected to Mission Planner using a USB cable so that firmware instillation, radio calibration, gain tuning, and log downloading can be completed while the airplane is on the ground. With the 3DR telemetry link mentioned before, the Mission Planner software allows the operator to turn any computer into a ground station. The aircraft's real time attitude, airspeed, location, flight mode and even the remaining battery life can all be monitored while the plane is in the air. The telemetry link also allows gain tuning to be done in flight and waypoints for autonomous missions can be added or edited using the Mission Planner software.

The ArduPlane software has a number of built in flight modes including: STABILIZE, FLY_BY_WIRE, LOITER, AUTO, and RETURN_TO_LAUNCH (RTL). The

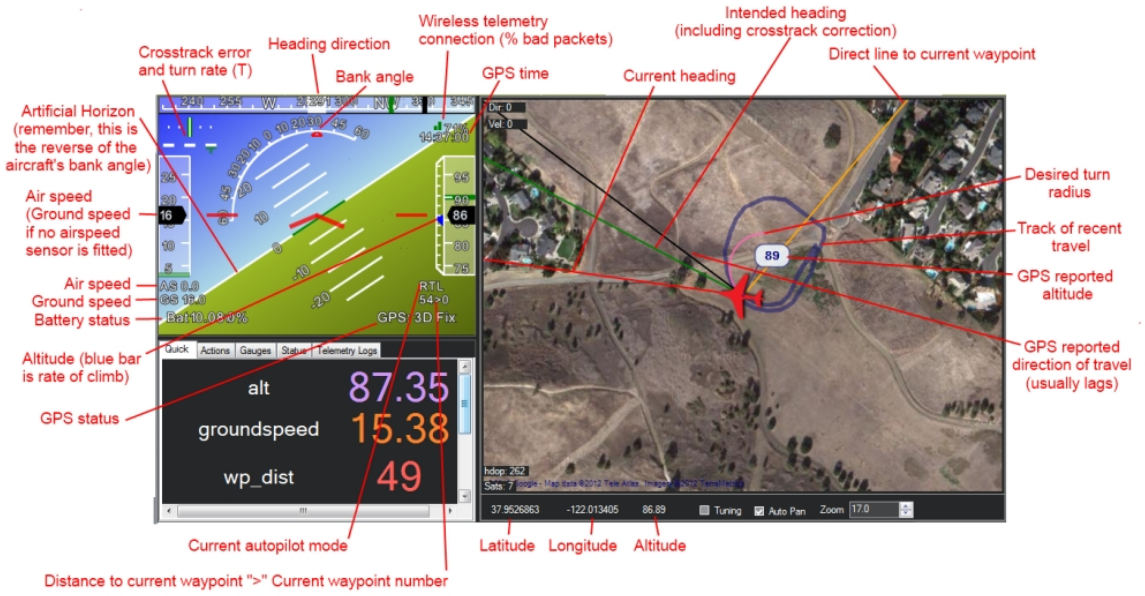


Figure 2.3: Mission Planner software and 3DR Telemetry Kit turn any computer into a UAV ground station.

fully autonomous flight modes (LOITER, AUTO, and RTL) use a pair of cascaded PID loops to control the aircraft. The outer loop uses a bank-to-turn controller to guide the aircraft to the next waypoint, while the inner loop stabilizes the aircraft at the attitude that is commanded by the outer loop. The stability augmentation modes (STABILIZE and FLY_BY_WIRE) only use the inner loop PID controllers to help stabilize the airplane but still allow the pilot to have general control of the aircraft. The aircraft's attitude is determined using quaternion based inertial navigation to avoid gimbal lock, however the inner loop stabilization is done based on Euler angle errors which causes problems when hovering. This problem, and the relationship between quaternions and Euler angles, is discussed extensively in Chapter 3.

In order to test both the PID and adaptive transition-to-hover controllers in an actual aircraft, three new stabilization flight modes were added to the standard ArduPlane software that attempt to hold a stable hover orientation instead of steady, level flight. The development of the new flight modes and the control logic that they use

is discussed in Chapter 4. It is important to note that only hover stabilization was attempted as part of this thesis, and that full waypoint based navigation while in hover orientation was out of the scope of this project.

Another particularly useful feature of the AdruPlane firmware and the Mission Planner software is that, together, they have built in functionality to allow hardware-in-the-loop (HIL) simulation through flight simulation programs like X-Plane and Flight Gear. While these programs' physics engines don't have the ability to accurately model the high angle of attack and post stall aerodynamics seen in transitional flight (and writing a new physics engine that was accurate enough and fast enough to run those types of simulations in real time would be another thesis in itself), the HIL simulations did provide an excellent method for checking basic functionality and control logic of the newly created flight modes.

Other recent works done at California Polytechnic State University, San Luis Obispo have also used the APM and other small autopilots as a standard platform for UAV research. Wallace [30] added potential function guidance capabilities to the Cloud Cap Piccolo autopilot for the potential use in autonomous obstacle avoidance. Lopez [19] then built upon the potential function guidance principle and used it to have multiple APM equipped UAVs fly in formation, using decentralized control techniques, that ran as a top level outer navigation control loop with the standard ArduPlane software controlling the flight path of each individual airplane. By using the APM and ArduPlane firmware, the research at Cal Poly has been able focus on adding capabilities to the autopilot rather than having to tackle the problem of developing and maintaining an autopilot from scratch. In addition, the simplified Arduino programing language and development support from DIY Drones help eliminate software development problems and keep the research work focus actual control theory instead of programming and hardware integration.

2.2 E-flite Carbon-Z Yak 54 Airframe

Since one of the goals of this thesis was to develop a transition-to-hover controller that did not need a specially designed airframe to be successful, it was important to choose a fairly common RC airframe to use in flight testing. The Carbon-Z Yak 54 [13], designed and manufactured by E-flite and shown in its stock configuration in Figure 2.4, was chosen for it's traditional single engine airframe configuration, as well as its power, ease of use, and its durability. The aircraft's design specifications are shown in Table 2.1.

Figure 2.4: E-flite Carbon-Z Yak 54

Wingspan	48.0 in.
Wing Area	525 sq. in.
Flying Weight	3.80 lbs.
Wing Loading	16.2 oz./sq. foot
Motor Type	Park 25 Brushless Outrunner
Motor Kv Rating	1000 Kv
Battery	4S 2800 mAh LiPo
Propeller Size	12 x 5.25

Table 2.1: Carbon-Z Yak 54 Specifications

The main requirement for the aircraft was that it has the ability to maintain stable hover under manual control without any modifications to the stock airframe or propulsion system. The Yak 54's brushless outrunner motor and 12 inch propeller give the aircraft a measured static thrust-to-weight ratio of 1.4 which is more than enough to maintain constant altitude and climb vertically when in hover orientation. Additionally, the aircraft was designed for aerobatics so the wings and stabilizers are extra stiff to help eliminate aeroelastic effects during high-G maneuvers (like the pull up at the beginning of a transition-to-hover maneuver), and the control surfaces and servos are over-sized to help increase control authority during low speed maneuvers like hovering and high angle of attack flying.

Figure 2.5: Autopilot and Airframe Integration

The second requirement for the test aircraft was that it was large enough to accommodate the addition of the APM system. Since the airplane has a stock flying weight of 3.80 pounds, the addition of the 0.155 pound APM system only results in a 4 percent increase in wing loading and had minimal effect on the airplane's aerodynamic performance. The flat surfaces under the airplane's plastic canopy provided the perfect mounting points for the APM board, GPS module, and telemetry antenna

as shown in Figure 2.5. The level surface helps with repeatability when the autopilot uses the accelerometers get an initial attitude solution, and mounting the autopilot very close to the vehicles center of gravity helps the accuracy of the inertial navigation system and minimizes potential effects on the longitudinal stability of the airplane.

Another interesting feature of the aircraft is that it is made using a special balsa/carbon fiber/EPP foam composite construction that makes the airframe light, but still very stiff and very durable. The extra durability of this airframe, compared to traditional balsa models was an added bonus, as test aircraft frequently suffer bumps and bruises during flight tests. The fact that replacement parts for the Yak 54 (even entire wings) are readily available on the E-flite website also helped reduce down time when damages occurred.

3 EULER ANGLES AND QUATERNIONS

The aircraft rigid body equations of motion are presented here along with details on how both quaternions and Euler angles were used in the control algorithm implementation. Regardless of the method chosen to describe the aircraft's orientation, the rigid body equations of motion are derived in Nelson [26] and described by

$$\vec{F} = m \left. \frac{\mathrm{d}\vec{v_c}}{\mathrm{d}t} \right|_B + m(\vec{\omega} \times \vec{v_c}) \tag{3.1}$$

$$\vec{M} = \left. \frac{\mathrm{d}\vec{H}}{\mathrm{d}t} \right|_B + \vec{\omega} \times \vec{H} \tag{3.2}$$

where $\vec{F}$ and $\vec{M}$ are the applied forces and moments in the inertial reference frame, m is total mass of the aircraft, $\vec{v_c}$ is the velocity of at the vehicle's center of mass, $\vec{\omega}$ is the angular velocity of the aircraft, $\vec{H}$ is the angular moment of the aircraft, and subscript B refers to the body reference frame of the aircraft. Details on how the aircraft's forces and moments were calculated for simulations are presented in Chapter 5.

A standard North-East-Down local inertial reference frame was used for all simulations and plots presented in this thesis. Since all flight tests took place over a short period of time and in an area of only a few acres, Coriolis effects due to the rotation of the earth were ignored and the inertial reference frame was fixed to an arbitrary point on the earth's surface near the takeoff location of the aircraft.

3.1 Euler Angle and Quaternion Definitions

Euler angles are traditionally used to represent aircraft attitude in aerospace literature because they are intuitive to use in control applications and are easy to visualize

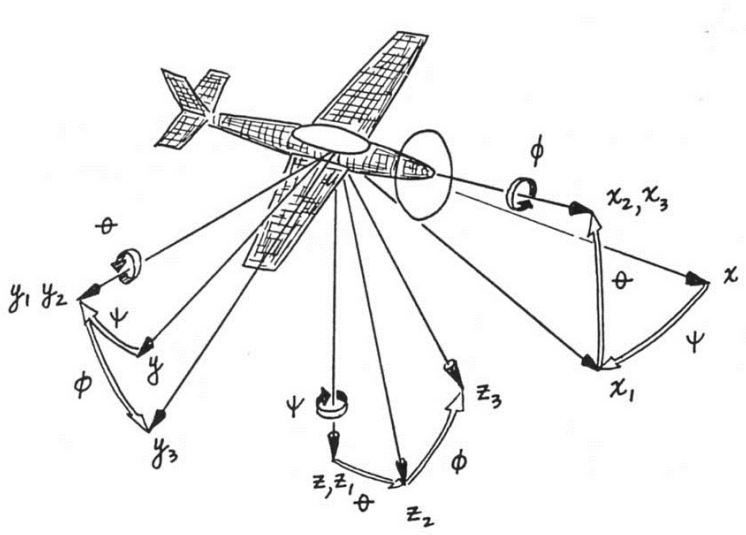


Figure 3.1: Euler Angle Attitude Representation

for pilots, but they have limitations due to singularities in their equations. Starting from an aircraft orientation with the nose facing North (the inertial frame x axis), the right wing facing East (the inertial frame y axis), and the belling facing down (the inertial frame z axis), each angle signifies a rotation about an individual axis. The order of rotations is non-commutative, so a standard order of rotations must be used. First the aircraft is rotated about the z axis by ψ, the yaw angle. Then the aircraft is rotated about the newly created y axis by the pitch angle, θ, and finally the aircraft is rotated about the new x axis by the roll angle, ϕ, as shown in Figure 3.1.

Quaternions, on the other hand, describe a vehicles attitude with respect to the inertial frame using four elements. The first three elements, called the vector portion of the quaternion and represented by $\vec{q}$, describe an axis of rotation while the fourth element specifies the magnitude of the rotation about that axis. A rotation of Θ

radians about the three-dimensional vector $\vec{v}$ is represented as the quaternion

$$q = \begin{pmatrix} \vec{q} \\ q_4 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} \sin\frac{\Theta}{2}v_1 \\ \sin\frac{\Theta}{2}v_2 \\ \sin\frac{\Theta}{2}v_2 \\ \cos\frac{\Theta}{2} \end{pmatrix}. \tag{3.3}$$

Any aircraft's attitude can be described using a single quaternion. This attitude quaternion represents the axis of rotation (defined in the inertial frame) and magnitude of the rotation to achieve the current aircraft orientation when starting from the nose North, right wing East, and belly down reference frame.

One important property of quaternions is that they actually contain redundant information. The four elements of any quaternion must create a unit vector, so given any combination of 3 elements of a valid quaternion, the fourth element can be easily determined.

3.2 Euler Angles and Quaternions in ArduPlane

As mentioned in Chapter 2, the ArduPlane inertial navigation system uses a quaternion based method for determining the current orientation of the aircraft, but then converts the quaternion orientation into Euler angles for use in the stabilization controllers. The advantage of using quaternion based inertial navigation is that quaternions are free of singularities while Euler angles are not. An Euler angle based scheme uses the matrix equation

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \tag{3.4}$$

to determine the Euler rates based on the body frame angular rates (p, q and r, also referred to as $\vec{\omega}$), and then integrating those rates to find the Euler angles. This

method is fine as long as the aircraft only flies at low pitch angles, however, if a pitch angle of $\theta = \pm\frac{\pi}{2}$ radians (± 90 degrees) is ever reached, the $\tan\theta$ term becomes undefined and causes a break down in the integration of the Euler rates commonly referred to as "gimbal lock." The quaternion rate equation, on the other hand, does not contain any singularities and is defined by

$$\begin{aligned} \dot{\vec{q}} &= \tfrac{1}{2}(q_4\vec{\omega} - \vec{\omega} \times \vec{q}), \\ \dot{q_4} &= -\tfrac{1}{2}\vec{\omega}^T\vec{q}. \end{aligned} \tag{3.5}$$

The reason that the quaternion orientation is converted to the singularity prone Euler angles is because the Euler angles are much easier to use for control implementation. The roll, pitch, and yaw angles of a conventional fixed-wing aircraft can be directly controlled by the ailerons, elevator, and rudder respectively. By simply taking the difference between the current angle and the desired angle for each of the rotation axes, a simple feedback loop can be established and individual control of each of the three Euler angles with minimal coupling can be achieved.

The problem with Euler angle difference based command is that does not work when the aircraft is in hover orientation for the same reason that Euler angle based navigation experiences gimbal lock. Say, for example, an airplane is flying due North and perfectly level. Its [roll, pitch, yaw] Euler angle vector would be represented by $[0, 0, 0]$ degrees. If the airplane was then commanded to rotate into hover orientation, $[0, 90, 0]$, the resulting Euler angle difference would be $[0, 90, 0]$ which if fed into aileron, elevator and rudder controller respectively, would only cause the elevator to deflect and the airplane to pitch up towards vertical. The problem comes when a wind disturbance or a controller overshoot causes the pitch angle to be greater than 90 degrees. When this happens, the Euler rates equations think the airplane is now traveling the opposite direction, which will cause unwanted control inputs for the roll and yaw directions. For example, say a wind gust pushes the aircraft to an orientation of $[0, 95, 0]$. Because of the nature of the Euler angle equations, the true Euler angle

orientation for that attitude is represented by $[-180, 85, -180]$ because based on the required yaw-pitch-roll rotation order, the way to get to that orientation would be to head due South, pull up to a pitch angle of 85 degrees and then roll the airplane belly up. Using simple Euler angle differences as control inputs when the aircraft is in this state would result in a control input vector of $[180, -5, 180]$ which would cause completely unnecessary (and rather large) deflections for both the ailerons and rudder and cause the airplane to diverge from its hover orientation.

There is, however, a simple work-around for this problem. By using a quaternion error to determine the most efficient route to the desired attitude and then converting that to an Euler angle rotation error, the Euler angle singularity can be avoided.

3.3 Quaternion Error Implementation

In order to avoid the problems that Euler angles present for an aircraft trying to maintain stable hover, modifications had to be made to the ArduPlane firmware that converted the all Euler angle based control scheme to a hybrid quaternion and Euler angle scheme. First, the aircraft's Euler angle orientation that is output by the inertial navigation code was converted to its quaternion equivalent using

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} \sin\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} - \cos\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\sin\frac{\psi}{2} - \sin\frac{\phi}{2}\sin\frac{\theta}{2}\cos\frac{\psi}{2} \\ \cos\frac{\phi}{2}\cos\frac{\theta}{2}\cos\frac{\psi}{2} + \sin\frac{\phi}{2}\sin\frac{\theta}{2}\sin\frac{\psi}{2} \end{pmatrix}. \tag{3.6}$$

The quaternion error is then calculated using the matrix equation

$$q_e = \begin{pmatrix} q_4 & q_3 & -q_2 & -q_1 \\ -q_3 & q_4 & q_1 & -q_2 \\ q_2 & -q_1 & q_4 & -q_3 \\ q_1 & q_2 & q_3 & q_4 \end{pmatrix} \begin{pmatrix} q_{c1} \\ q_{c2} \\ q_{c3} \\ q_{c4} \end{pmatrix}. \tag{3.7}$$

The quaternion error, q_e is defined as the quaternion rotation needed to go from the aircraft's current attitude, q, to the commanded quaternion attitude, q_c, which is equal to $[0, \pi/2, 0, \pi/2]$ for a due North, perfectly vertical hover attitude. Finally, the quaternion error is then converted back into Euler angles using the equation

$$\begin{pmatrix} \phi_e \\ \theta_e \\ \psi_e \end{pmatrix} = \begin{pmatrix} \tan^{-1} \frac{2(q_{e2}q_{e3}+q_{e4}q_{e1})}{1-2({q_{e_1}}^2+{q_{e_2}}^2)} \\ \sin^{-1}\left(-2(q_{e1}q_{e3}-q_{e4}q_{e2})\right) \\ \tan^{-1} \frac{2(q_{e1}q_{e2}+q_{e4}q_{e3})}{1-2({q_{e_2}}^2+{q_{e_3}}^2)} \end{pmatrix} \tag{3.8}$$

and the resulting Euler angle rotation error can be used as the control input without the problems discussed in the previous section.

The reason this method works is because the quaternion error is actually a rotation and not just a simple difference, so the resulting Euler angle error is also a rotation with respect to the current attitude instead of the *difference* between the current attitude and desired attitude with respect to the inertial frame. If this quaternion error method is applied to the example presented in Section 3.2, the resulting Euler angle error to go from a current Euler angle attitude of $[-180, 85, -180]$ degrees to the desired attitude of $[0, 90, 0]$, is equal to $[0, -5, 0]$ since the aircraft only needs to decrease its pitch angle by 5 degrees *relative to its current attitude* in order for it to return to the desired hover orientation.

The one problem with this method is that converting the quaternion error back to Euler angles reintroduces a singularity into the aircrafts attitude. Now, however, the singularity is only a problem when the *rotation* needed to go from the current attitude to the desired attitude contains a pitch rotation greater than or equal to 90 degrees. This would happen, for example, if the current pitch angle was -10 degrees and the desired pitch angle was 90 degrees, with respect to the inertial frame. Unlike the problems with Euler angle differences discussed in Section 3.2, however, this problem can be easily avoided by using a logic gate to command a lesser pitch angle change until the aircraft's attitude gets within 90 degrees of its final target. For

example, in the case above, the desired pitch angle could be set to 30 degrees until the current pitch angle crossed zero at which point it would be changed to 90 degrees to obtain the desired final results. This logic based method can't be used with the Euler angle differences though, because the point where the singularity occurs *is* the desired attitude when attempting to hover, as opposed to the singularity occurring only for a short period of time during the rotation to the desired attitude when using the quaternion error method.

While the quaternion error could technically be used directly as a control input, like it is often done in spacecraft, its lack of physical meaning in the context of aileron/elevator/rudder control would make its integration into the standard ArduPlane firmware difficult. Gain tuning would become much less intuitive and would probably require different sets of PID gains to be used for hover flight modes which use the quaternion error, compared to the other level flight modes that would still use the Euler angle difference. Since ease of use was one of the goals of this thesis, it was decided that it would be better to use the quaternion based Euler angle rotation error as the final control input, instead of the quaternion error, and simply handle the singularity problems using the logic gate method with the hope of reducing the need for major changes to the AdruPlane firmware.

4 CONTROL ARCHITECTURE

The standard ArduPlane firmware uses a pair of cascading PID control loops in order to control the aircraft in fully autonomous flight modes. The outer loop controllers use GPS waypoints to calculate the aircraft's desired attitude and airspeed and then the inner loop controllers convert the attitude and airspeed errors into actual control surface deflections and throttle settings. In pilot assisted flight modes, such as STABILIZE or FLY_BY_WIRE, only the inner loop PID controllers are used. The pilot's transmitter inputs are used to generate the desired aircraft attitude and then the inner loop controllers are used to generate the necessary control surface deflections while the throttle is typically controlled by the pilot directly.

This thesis focuses only on the inner loop attitude stabilization of the UAV during the transition-to-hover maneuvers so all of the controller modification were done at the inner loop level while the outer loop is left unchanged and unused when using the "hover" flight modes. This means that the aircraft's speed and position are never controlled directly during the transition or while the airplane is hovering. Doing so would require heavy modification of the outer loop navigation controllers which is out of the scope of this project, but is discussed as possible future work in Section 8.1.

Three new APM flight modes were created, each with intent of transitioning the airplane from level flight to hover flight, and then maintaining hover flight, but each with a slightly different control logic. These control modes will be referred to throughout the rest of this report as HOVER_PID_STEP, HOVER_PID_REF, and HOVER_ADAPTIVE. The HOVER_PID_STEP and HOVER_PID_REF flight modes both use the ArduPlane PID controllers described in Section 4.1.1 to control the roll, pitch and yaw of the aircraft, but the first tries to complete the transition as fast as

possible by tracking a step input, while the second attempts to track a second order reference model to complete the transition. HOVER_ADAPTIVE, on the other hand, uses Model Reference Adaptive Control (MRAC) to track the same reference models used by the HOVER_PID_REF flight mode. Detailed descriptions of the ArduPlane PID controllers, the MRAC theory, and the design of the reference models are all included in this chapter, followed by a summary of the final controller gains and control logic used for each of the three new hover flight modes.

4.1 Proportional, Integral, Derivative Control

A classical Proportional, Integral, Derivative (PID) controller has the frequency domain form

$$u(s) = k_p e(s) + \frac{k_i}{s} e(s) + s k_d e(s) \tag{4.1}$$

which results in the time domain solution

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{d}{dt} e(t) \tag{4.2}$$

where e is the error that is used as the controller input, k_p is the proportional gain, k_i is the integral gain, k_d is the derivative gain, and u is the controller output.

ArduPlane uses three separate PID controllers, all with the same format but with individually tuned gains, to control the roll, pitch and yaw axes of the aircraft individually. The standard APM method for generating the input error (e) for each axis is to simply take the difference between the desired Euler angles and the current aircraft Euler angles. As discussed in Chapter 3, this method works fine when in level flight but breaks down due to a singularity in the Euler angle math when the pitch angle reaches 90 degrees. The quaternion error method discussed in Section 3.3 is used to work around this problem and generate the "hover safe" roll, pitch and yaw errors (in degrees), which are then fed into the PID controllers.

4.1.1 ArduPlane PID Control

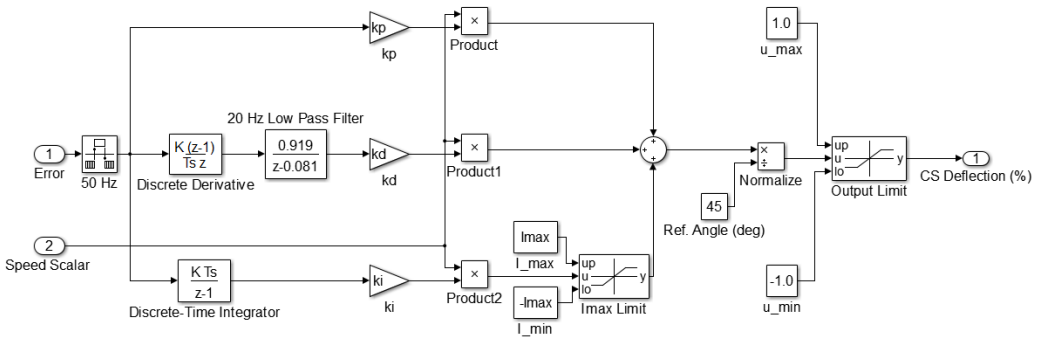


Figure 4.1: ArduPlane PID Controller Block Diagram

In addition to the quaternion error modification discussed in Section 3.3, ArduPlane also uses a slightly modified version of a classical PID controller, shown in block diagram form in Figure 4.1, to account for some of the challenges faced when controlling a fixed-wing aircraft. First, in addition to the proportional, integral and derivative control gains, the ArduPlane PID controller introduces another gain, k_{ss}, which is referred to as a "speed scalar" so that the transfer function becomes

$$u(s) = k_{ss}k_p e(s) + \frac{k_{ss}k_i}{s}e(s) + sk_{ss}k_d e(s). \tag{4.3}$$

The speed scalar is determined by dividing a reference aircraft ground speed (which has a default value of 50 ft/sec) by the current ground speed such that

$$k_{ss} = \frac{50ft/sec}{V_{ground}}, \tag{4.4}$$

but is limited to the range $0.5 \leq k_{ss} \leq 2.0$. The speed scalar helps account for the fact that control surface deflections are less effective at low speeds than they are at high speeds, by modifying the magnitudes of the P, I, and D control gains. The ground speed is calculated by taking a derivative of the GPS ground track and is used instead of airspeed because an airspeed probe, while available as an optional add-on, is not a

part of the standard ArduPilot package and was not used as part of this thesis. The reference ground speed may be easily changed in the firmware but was left at the default value of 50 ft/sec to ensure consistency during flight tests.

The second modification the APM PID controller uses is that it constrains the magnitude of the integral term to be within a user defined value, I_{max}, such that

$$-I_{max} \leq k_{ss} k_i \int_0^t e(\tau) d\tau \leq I_{max} \tag{4.5}$$

in the time domain solution. Limiting the integral term helps prevent integral windup in the event of strong wind gusts or if the pilot decides to manually override the autopilot control (a feature that is built into all ArduPilot flight modes) for an extended period of time while the PID controller still runs in the background. The default value for I_{max} is 5.0 but this value had to be modified for each control axis to achieve the desired performance. The final chosen values for each of the PID controller gains are shown later in this chapter in Table 4.1.

The third APM PID controller modification is the addition of a low-pass filter to the error derivative term before being multiplied with the derivative gain and speed scalar. Since the APM attitude calculations are run at 50 Hz, the low pass filter is designed with a 20 Hz cutoff frequency to help reduce the effects of sensor noise from the gyros and accelerometers on the error derivative calculation. The low pass filter is implemented discretely in the firmware using the equation

$$\left[\frac{de(t)}{dt}\right]_i = \left[\frac{de(t)}{dt}\right]_{i-2} + \frac{\Delta t}{RC + \Delta t}\left(\left[\frac{de(t)}{dt}\right]_{i-1} - \left[\frac{de(t)}{dt}\right]_{i-2}\right) \tag{4.6}$$

where

$$RC = \frac{1}{2\pi * 20 \text{ Hz}} \tag{4.7}$$

and i is the current time step, and Δt is the time between time steps which is not always constant since the APM always tries to run at a maximum of 50 Hz but can sometimes get slowed down depending on how many calculations it is doing.

The final APM PID controller modification compared to the classical PID showed in Equation 4.2 has to do with the units needed for the controller output. The APM PID controllers all operate on Euler errors that are expressed in degrees, but the autopilot board has to send a Pulse Width Modulation (PWM) signal to the servos in order to move the control surfaces to the desired deflection angle. To do this the sum of the resulting proportional, integral and derivative components is first divided by 45 degrees (the value of which was arbitrarily chosen by the APM designers) to effectively normalize the control output. The resulting number is then limited to a minimum value of -1.0 and a maximum value of +1.0 so that the commanded control surface deflections are limited to $\pm 100\%$. The resulting ratio between ± 1 is then converted to a PWM rate based on the minimum, maximum, and trim PWM rates that are recorded when performing the initial transmitter calibration that is required before the autopilot can ever be used in flight.

4.1.2 PID Gain Tuning

Control gain tuning for PID controllers is typically done using programs like Matlab and Simulink in order to ensure that the final controller design meets performance specifications like rise time, maximum overshoot and settling time requirements. However, detailed knowledge of the plant is needed in order to do this type of analytical tuning. While this type of system identification and PID tuning has been used in previous transition-to-hover works, such as [14], to achieve good transitional and hover flight performance, the ArduPilot and ArduPlane firmware were designed for "plug-and-play" style use with many different types of airframes without the need for advanced knowledge of the system dynamics. To honor the spirit of the ArduPilot project, and to hopefully make the newly created flight modes easy to use for the rest of the DIY Drones community, all gain tuning of the ArduPlane PID controllers was done based on flight tests and simulated flights using the trial and error method sug-

gested on the DIY Drones help site [12], which all ArduPilot users are encouraged to reference and is available to the general public. No system identification or analytical gain tuning methods were used to determine the any of the control gains used in the flights or simulations conducted for this thesis.

Thanks to the Mission Planner software and 3DR telemetry link that was installed on the plane, gain tuning could be done while the aircraft was still flying, that way the effects of gain changes could be seen immediately and compared to the original controller performance. The DIY Drones suggested gain tuning method is comprised of a few easy to follow steps, which are to be performed for each of the roll, pitch and yaw axes individually:

1. Before flight, set the derivative and integral gains for the desired axis to zero and set the proportional gain for that axis to a relatively low value like 0.10.

2. Once airborne and at a safe altitude, switch the autopilot into the STABILIZE mode and use the transmitter to introduce small, quick disturbances to the control axis you are trying to tune and watch the response of the aircraft as it returns back to steady level flight.

3. Increase or decrease the proportional gain for the selected controller as needed until the aircraft responds to the pilot initiated disturbances quickly but with minimal overshoot and oscillations. It is acceptable for the airplane to not fully return to its initial attitude at this point so only increase the proportional gain to the point where a crisp response is created. The steady state error is addressed in the next step.

4. Increase the integral gain for the selected controller until the aircraft is able to recover from the user initiated disturbances and eliminates the steady state error in a timely manor (i.e. 0 degrees roll/pitch angle, or returns to the original

heading within one or two seconds). Typically the integrator gain value is no more than half the magnitude of the proportional gain. If the aircraft is never fully able to eliminate the steady state error is it because the default maximum integrator value of 5.0 is too low. Increase the I_{max} value until the steady state error is eliminated completely.

5. Add small amounts of derivative gain to help smooth the control response and eliminate overshoot. Be cautious when adding derivative gain because too much can hurt the performance of the system since the discrete derivatives of the Euler angle errors can be affected by sensor noise. Typically a derivative gain value about 1/10 the magnitude of the proportional gain is sufficient.

During initial testing, this process was used to establish separate sets of control gains for level flight and hover flight. The reason for this was because the lack of airflow over the control surfaces during hover flight meant that the control surfaces needed much larger deflections to maintain the desired hover attitude compared to the deflections needed to maintain high speed level flight. While these separate sets of control gains did work well for their respective uses, they made using both level flight modes and hover flight modes in the same flight nearly impossible because uploading the lower magnitude flight gains meant that the controllers would not have enough control authority to maintain stable hover, while using the larger hover flight gains would cause large overshoot and oscillation problems when attempting to use the level flight modes like STABILIZE and FLY_BY_WIRE. As mentioned in the introduction, one of he main purposes of having a fixed-wing airplane that can hover is to take advantage of the increased speed and endurance while still having hover capabilities, so having an autopilot that can only autonomously control the airplane in either level or hover flight, and not both, defeats the purpose.

The initial answer to solving the problem of needing two sets of control gains

was to modify the ArduPlane firmware to have two instances of the PID controllers running: one with the smaller level flight gains, and one with the larger hover flight gains. These two PID controller sets would then hand off control of the airplane to one another depending on the current flight mode and aircraft orientation. This method would have required heavy modification of the ArduPlane firmware and would make the autopilot and Mission Planner setup process very difficult for normal users. Instead, a much simpler solution was found that produced comparable results to having two unique sets of control gains.

Upon further examination of the differences between the level flight gains and hover flight gains, it was discovered that the hover flight gains were all approximately proportional to their corresponding level flight gain values by the same scalar value. In other words, the proportional, integral and derivative gains used in hover flight were all about 5 times larger than their corresponding level flight gains. This made sense since the same tuning process was used to establish both sets of control gains. Since the speed scalar modification was already built in to the PID controllers to increase control surface deflections at lower speeds, it was easy to manually override the speed scalar value when the aircraft is hovering. The firmware was modified to increase the speed scalar to a value of 5.0, instead of its original maximum of 2.0 (as detailed in Section 4.1.1), once the aircraft reaches hover orientation (defined through testing as pitch $\geq$ 85 degrees). This modification made it so that the HOVER_PID_STEP and HOVER_PID_REF modes would start the transition-to-hover using the level flight control gains. As the airplane pitches up and slows down the speed scalar would gradually increase from 0.5 to 2.0. Then, once the pitch angle exceeded 85 degrees, the speed scalar would increase to 5.0 which gave the same effect as using the separately tuned set of hover flight control gains when hovering.

Using this method was especially effective for maintaining ease of use because it only requires that the pilot establish a single set of level flight gains when doing

the initial autopilot setup and calibration. Determining a unique set of hover flight gains that were capable of maintaining stable hover proved to be one of the most difficult parts of this thesis and required many flight tests to even reach a point where tuning for improved performance, instead of just tuning for successful hover, could even occur. In the event that another ArduPilot user attempts to use the new hover capabilities in a different airframe, the speed scalar override method will allow initial setup to only require establishing a set of level flight gains and then manually increasing the value of the hover speed scalar until stable hover can be maintained. Once that happens, small changes to the level flight gains can be made to improve both hover and level flight performance.

Level Flight Control Gains				
Control Axis	k_p	k_i	k_d	I_{max}
Roll	0.15	0.02	0.01	10.0
Pitch	0.27	0.02	0.01	20.0
Yaw	0.35	0.04	0.01	20.0

Table 4.1: Level Flight PID Control Gains

Hover Flight Control Gains = Level Gains $*$ 5.0				
Control Axis	$k_{ss} * k_p$	$k_{ss} * k_i$	$k_{ss} * k_d$	I_{max}
Roll	0.75	0.10	0.05	10.0
Pitch	1.35	0.10	0.05	20.0
Yaw	1.75	0.20	0.05	20.0

Table 4.2: Hover Flight PID Control Gains

The final values of the level flight control gains used for the Carbon-Z Yak 54 airframe during all simulations and flight tests discussed in the results chapters of this thesis are presented in Table 4.1, and the equivalent hover flight control gains are presented in Table 4.2. It is important to note that only the proportional, integral and derivative gain values are affected by the speed scalar override. The integrator maximum values remain unchanged when in hover orientation.

4.1.3 Throttle Controller

The standard ArduPlane control logic uses two separate methods for determining the throttle setting depending on whether an airspeed probe is installed or not. When the autopilot is not equipped with an airspeed probe, the throttle is controlled by a simple feed forward control loop that is dependent on a user specified "cruise" throttle setting (default value of 45%) and a feed forward gain (k_{ff}) that operates on the elevator input such that

$$u_{throttle} = 45 + k_{ff} * u_{elevator} \tag{4.8}$$

to command a throttle setting between 0% and 100%. This gives the effect that the airplane throttles down when it is diving and throttles up when it is climbing to hopefully maintain a semi-constant airspeed.

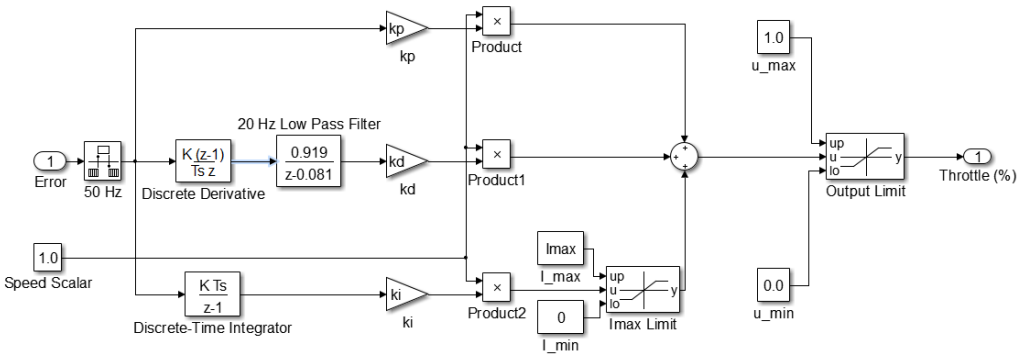


Figure 4.2: ArduPlane Throttle PID Controller Block Diagram

When an airspeed probe is installed, however, the throttle is controlled by the same type of PID controller that performs the attitude stabilization except for a few small differences. First, the input error is generated by calculating a total energy error (kinetic + potential energy) based on the desired airspeed and altitude. Secondly, the speed scalar is forced to equal 1.0 regardless of the speed of the aircraft, and the integrator term is limited to a minimum value of 0 instead of $-I_{max}$. Lastly, the

resulting sum of the proportional, integral and derivative components is not divided by 45 before being limited to a final range between 0 and 1. A block diagram of the resulting throttle PID controller is shown in Figure 4.2.

As mentioned previously, an airspeed sensor was not equipped on the ArduPilot board used in this thesis, because it is not part of the standard ArduPilot package. Additionally, the low cost airspeed probes that are available for easy plug-and-play use with the ArduPilot board are inaccurate when used at the high angle of attack and at the extremely low airspeeds experienced when the aircraft is hovering. So instead of using total energy error as the throttle PID controller input, descent/climb rate was used so that only an altitude measurement was needed to generate the throttle setting. This modification was relatively easy to make since the APM comes equipped with both a barometric pressure sensor and a GPS unit capable of measuring altitude as part of the standard package.

Originally, the throttle controller was going to be modified to control the altitude of the airplane directly while it is in the hover flight modes. The initial altitude of the aircraft would be saved when the transition is first initiated and then the throttle would be controlled by the PID controller to try and maintain that same altitude throughout the transition and while hovering. This method was flawed, though, because during faster transitions it was common for the aircraft to gain altitude sometimes in excess of 100 feet, so once stable hover was establish the throttle controller would have to try to slowly lower the aircraft back down to the original altitude, often times hindering the precision of the hover, rather than just maintaining the altitude the aircraft was at when it started hovering.

To solve this problem, the throttle PID controller was set up to use descent/climb rate as the control input. The descent/climb rate is calculated simply by taking the derivative of the mix altitude (a combination of the altitude measurements from the

barometer and the GPS unit) and then running it through the same 20 Hz low pass filter in Equation 4.6 to help eliminate high frequency sensor noise. When in hover flight modes, the throttle stick on the transmitter is changed to a fly-by-wire style input so that the pilot commands a descent/climb rate between ±6.5 feet per second (±2 m/sec). The value of 6.5 feet per second was determined through simulation and flight testing to be the maximum controllable descent/climb rate the airplane could handle such that if the airplane were to fall or climb any faster than that it would likely lose all control effectiveness and diverge from stable hover. Using this control input method, the pilot can simply place the throttle stick on the transmitter to half throttle in order to command a descent rate of 0 feet per second which means that the aircraft will just try to maintain whatever altitude it is currently at. That way the airplane is only trying to climb or descend when hovering if the pilot commands it to, and it will be doing so at the user commanded rate instead of trying to reach a final desired altitude as fast as possible.

Since the throttle PID controller was modified to work using descent rate error instead of total energy error, the resulting PID control gains were very different than they would be for level flight flight when an airspeed sensor is installed. However, without the airspeed sensor installed the default ArduPlane flight modes will always used the feed forward method described in Equation 4.8, or use a simple manual pass through for the throttle, so none of the other standard flight modes' functionality was affected. The final control gains for the modified descent/climb rate to throttle PID controller are shown in Table 4.3. These gains, with the modified throttle PID controller, are used throughout transition and hover and no changes are made to the speed scalar value of 1.0 based on the speed or attitude of the aircraft. Since the control output is not normalized for the throttle controller like it is for the control surfaces, the I_{max} value of 90.0 literally means that the integrator term can command up to 90% throttle on its own. This is important for being able to eliminate steady

state error since the propeller usually needs at least 65% throttle to produce enough thrust to offset the aircraft's weight.

Hover Flight Throttle Control Gains				
Control Type	k_p	k_i	k_d	I_{max}
Throttle	0.60	0.10	0.01	90.0

Table 4.3: Hover Flight Throttle PID Control Gains

After initial simulations and flight testing, it became apparent that only using the descent/climb rate based PID controller to handle the throttle controls would not be enough to maintain stable hover in the presence of wind disturbances. While the PID controller was effective at maintaining the desired vertical velocity, the relatively low amount of airflow over the control surfaces meant that the control surfaces were sometimes not able to counteract strong wind gusts and keep the aircraft from losing hover orientation while the throttle was at the approximate 65% needed to maintain thrust equal to weight. When an RC pilot manually hovers a fixed wing aircraft and experiences a wind disturbance, he or she has to carefully, but quickly, increase the throttle in order to first create extra airflow over the control surfaces to increase control effectiveness, and then decrease the throttle just enough to stop the upward acceleration but still maintain control authority and stability. This control sequence was replicated by creating a hover divergence logic controller to supplement the throttle PID controller and help the aircraft maintain stable hover. The hover divergence logic controller works by setting the divergence throttle to 75% if either the the magnitude of the pitch or yaw angle errors exceeds 5 degrees or by setting the throttle to 0% if the limit is not exceeded. It is shown as part of the complete hover throttle controller logic in Figure 4.3.

When in one of the hover flight modes, both the hover divergence logic controller and the throttle PID controller run simultaneously and each generate a suggested throttle setting. The throttle setting is then chosen by taking the maximum of the

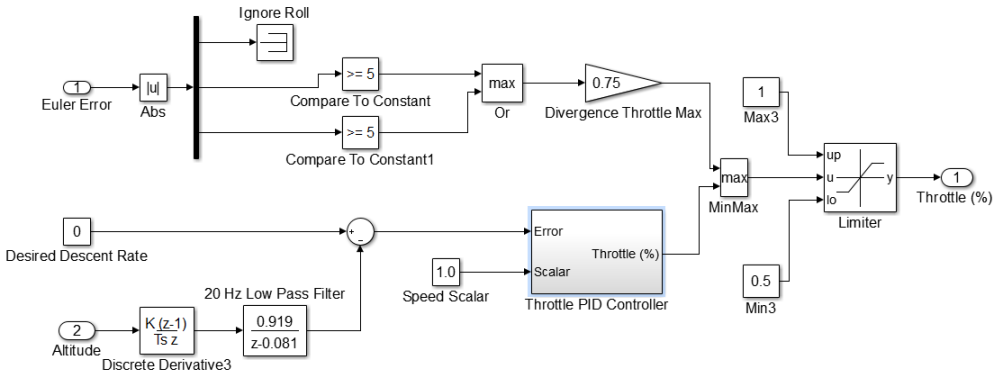


Figure 4.3: Hover Throttle Control Logic Block Diagram

two suggestions in order to ensure that both the controllers are satisfied. For example, if the airplane needs 65% throttle to maintain a constant altitude but has a pitch error of 7 degrees, the throttle will be set to 75% because it is more than is needed to maintain altitude but also sufficient to help return the aircraft back to stable hover. If instead a climb rate of 5 feet per second is commanded, which requires 90% throttle to achieve, and the pitch error was still 7 degrees, the throttle would be set to 90% because it is more than what is needed to help return the airplane back to stable hover but also sufficient to maintain the desired climb rate. Finally, the minimum throttle is set to 50% any time the ArduPilot is in one of the three hover flight modes to ensure there is always some airflow over the control surfaces during transition and hover, but is still low enough to allow the aircraft to bleed off speed and descend when necessary.

It is important to note that the roll error is ignored in the hover divergence logic controller because torque roll (due to propeller and motor acceleration) plays a major part in the roll dynamics of the aircraft, so increasing the throttle in an attempt to help reduce roll error may actually make the problem worse and not better. As long as the nose of the aircraft is still pointed straight in the air (minimal yaw and

pitch error), the aircraft is considered to be in a stable hover even if the aircraft is rotated along its roll axis a large amount. While it may take a little time, the attitude stabilization controllers will eventually be able to correct the roll error without the need for additional airflow over the ailerons since there is no immediate threat to the stability of the hover.

Finally, even though a Model Reference Adaptive Controller is used to control the attitude stabilization of the aircraft when in the HOVER_ADAPTIVE flight mode, the same combination of PID and logic based controllers are used to determine the throttle setting when in any of the three hover flight modes. The throttle PID controller was not replaced with an adaptive controller when in HOVER_ADAPTIVE mode because initial testing showed that MRAC was not a good fit for the throttle controller due to the way the adaptive algorithm works. Details on the MRAC theory are presented in Section 4.2, but the result of using the adaptive control to generate a throttle command was essentially a bang-bang controller which set the throttle to either the minimum or maximum allowed value. While this control method did not make hovering completely impossible, the large changes in throttle did amplify the torque roll problems already seen during PID control testing, and also caused the airplane to undergo heavy oscillations in altitude that would occasionally cause large negative vertical velocities and make the aircraft diverge from stable hover. The PID and logic based throttle control method discussed in this section was simply a more natural and more effective way of controlling the aircraft's throttle during the transition maneuver regardless of the method used to control the aircraft's attitude.

4.2 Model Reference Adaptive Control

This section discusses the theory of the Model Reference Adaptive Controller and how it was implemented in the HOVER_ADAPTIVE flight mode. This specific

formulation of the MRAC is sometimes also called Adaptive Output Feedback (AOF) control and follows the derivation and stability proofs found in [22] and [28]. A model of the system is shown in Figure 4.4.

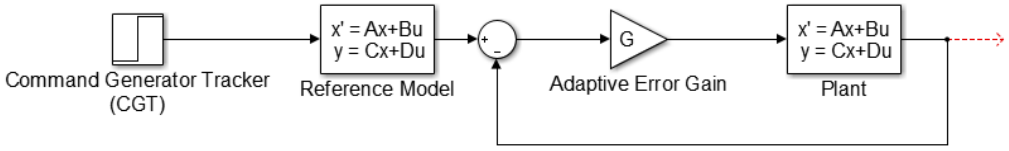


Figure 4.4: Simple Nonlinear Model Reference Adaptive Controller

4.2.1 MRAC Setup

Consider the N^{th} order nonlinear system represented as

$$\begin{aligned} \dot{\underline{x}}_p &= A_p \underline{x}_p + B_p \underline{u}_p + f(\underline{x}_p) \\ \underline{y}_p &= C_p \underline{x}_p \end{aligned} \tag{4.9}$$

where the subscript p denotes a property of the plant, and $f(\underline{x}_p)$ represents the nonlinear portion of the system. Also, $A_p \in \Re^{NxN}$ and u_p, $y_p \in \Re^N$ (i.e. the system is square. The system is considered to be output feed back stable if there exists some G_e^* such that the control law $\underline{u}_p = G_e^* \underline{y}_p$ where the system $(A_p + B_p G_e^* C_p)$ is exponentially stable.

Now consider the M^{th} order reference model described by

$$\begin{aligned} \dot{\underline{x}}_m &= A_m \underline{x}_m + B_m \underline{u}_m + g(\underline{x}_m) \\ \underline{y}_m &= C_m \underline{x}_m \end{aligned} \tag{4.10}$$

where the subscript m denotes a property of the reference model, and $g(\underline{x}_m)$ represents the nonlinear portion of the reference model. Also, $A_m \in \Re^{MxM}$ but it is important to note that the size of the reference model is independent of the size of the system, which means that it can be made lower order than the system in Equation 4.9 if

desired. Next the Command Generator Tracker (CGT) is used to determine the inputs for the reference model and is defined by

$$\begin{aligned} \dot{\underline{x}}_q &= A_q \underline{x}_q \\ \underline{y}_q &= C_q \underline{x}_q \end{aligned} \tag{4.11}$$

where the subscript q denotes a property of the CGT, and the only requirement for the CGT is that it is stable. All flights and simulations discussed in this thesis were performed with a simple step function as the CGT, where the desired pitch instantaneously increases from 0 degrees to 90 degrees, while heading is held constant and roll is desired to be zero throughout the transition and hover. The different reference models that were used are discussed in Section 4.3.

Next, consider an ideal system, which is the same order as the plant and defined by

$$\begin{aligned} \dot{\underline{x}}_* &= A_* \underline{x}_* + B_* \underline{u}_* + f(\underline{x}_*) \\ \underline{y}_* &= C_* \underline{x}_* \end{aligned} \tag{4.12}$$

where the subscript $*$ denotes a property of the ideal system, and $f(\underline{x}_*)$ represents the nonlinear portion of the ideal system. This system represents the ideal trajectories of the plant. The ideal system theoretically tracks the reference model perfectly such that $\underline{y}_m = \underline{y}_*$ for all $t > 0$. If the ideal system tracks the reference model for all $t > 0$ then that implies that the reference model can be made of some linear combination of the ideal system such that

$$\begin{bmatrix} \underline{x}_* \\ \underline{u}_* \end{bmatrix} = \begin{bmatrix} S^*_{11} & S^*_{12} \\ S^*_{21} & S^*_{22} \end{bmatrix} \begin{bmatrix} \underline{x}_m \\ \underline{u}_m \end{bmatrix} \tag{4.13}$$

If the derivative of Equation 4.13 is taken the results are substituted into Equations 4.9, 4.10, 4.11 and 4.12, solving for like terms gives that the following conditions must

be true:

$$\begin{aligned} A_p S_{11}^* + B_p S_{21}^* &= S_{11}^* A_m \\ (A_p S_{12}^* + B_p S_{22}^*) C_q &= S_{11}^* B_m C_q + S_{12}^* C_q A_q \\ f(\underline{x}_*) &= S_{11}^* g(\underline{x}_m) \\ C_p S_{11}^* &= C_m \\ C_p S_{12}^* C_q &= 0. \end{aligned} \tag{4.14}$$

These conditions are called the matching conditions and if the plant, reference model, and CGT satisfy the matching conditions for some combination of S_{ij}^* then the system is said to be *totally consistent*. Additionally, if the input to the ideal system, $\underline{u}_*$, is assumed to *not* be a function of $\underline{x}_m$ or $\underline{u}_m$, then $S_{21}^* = S_{22}^* = 0$ and the matching conditions simplify to

$$\begin{aligned} A_p S_{11}^* &= S_{11}^* A_m \\ A_p S_{12}^* C_q &= S_{11}^* B_m C_q + S_{12}^* C_q A_q \\ f(\underline{x}_*) &= S_{11}^* g(\underline{x}_m) \\ C_p S_{11}^* &= C_m \\ C_p S_{12}^* C_q &= 0. \end{aligned} \tag{4.15}$$

Next the output error and the state error of the system are defined, respectively, as

$$\underline{e}_y \equiv f(\underline{y}_p, \underline{y}_m) = f(\underline{E}_p, \underline{E}_m) \tag{4.16}$$

$$\underline{e} \equiv f(\underline{x}_p, \underline{x}_*) = f(\underline{E}_p, \underline{E}_*). \tag{4.17}$$

where $f(\underline{y}_p, \underline{y}_m)$ and $f(\underline{x}_p, \underline{x}_*)$ are calculated using the Euler rotation error method described in Section 3.3. Since the plant and reference model are fully observable in this application (i.e. $C_p = C_m = C_* = I$), $\underline{y}_p$ and $\underline{y}_m$ are equal to the states of the plant and reference model respectively, which are described using the Euler angle sets $\underline{E}_p$ and $\underline{E}_m$.

It is important to note the defining the state and output errors using the using the Euler rotation error method from Section 3.3 does change the traditional MRAC

stability proof from [22] and [28] slightly. [22] and [28] both used MRAC to control spacecraft and operated on the spacecraft's quaternion vector directly, which resulted in the output and state errors being defined as

$$\underline{e}_y \equiv \underline{y}_p - \underline{y}_m = \vec{q_p} - \vec{q_m} \tag{4.18}$$

$$\underline{e} \equiv \underline{x}_p - \underline{x}_* = \vec{q_p} - \vec{q_*}. \tag{4.19}$$

Using the quaternion vector instead of the Euler angles to describe the vehicle's attitude does not change the proof at all because the system is still 3x3 square, but defining the errors using simple differences instead of quaternion rotation errors definitely does. Simplifications that take place in the traditional stability proof are dependent on the fact that the state error is calculated using a simple difference and break down when the definition is changed to the rotation error that is calculated using Equation 3.7. Implementing the adaptive controller using the quaternion vector difference method was attempted as part of this thesis, but was very ineffective because of the lack of physical meaning in the quaternion vector as described in Chapter 3.

It is interesting then that the MRAC method was still able to effectively control the airplane with the state and output errors defined as Euler rotation errors even though that meant the mathematical stability proof was no longer valid. One explanation for this might be that the Euler rotation error is approximately equal to the simple Euler angle difference when the errors are small (less than 10 degrees) for each of the three axes. This condition is sometimes met throughout the transition when the adaptive controller is doing a good job of tracking the input reference model, so Equation 4.16 can be rewritten as

$$\underline{e}_y \equiv f(\underline{y}_p, \underline{y}_m) \approx \underline{y}_p - \underline{y}_m \tag{4.20}$$

$$\underline{e} \equiv f(\underline{x}_p, \underline{x}_*) \approx \underline{x}_p - \underline{x}_* \tag{4.21}$$

as long as the Euler errors are small. This small error assumption essentially linearizes the system so that it matches the definitions in [22] and [28] where the stability proof is valid. However, stability of the linearized system does not guarantee stability of the full non-linear system and more work would need to be done to prove stability in this way. Simulation and flight tests results, discussed in detail in Chapters 6 and 7, showed that the MRAC algorithm often broke down when any of the Euler angle errors became too large, so it is possible that the system is only mathematically stable as long as the errors can be held to low values.

It was also noticed that the MRAC stability proof is only dependent on the state error being defined as a difference, but not the output error. It is possible then that the errors could be defined independently as

$$\underline{e}_y \equiv f(\underline{y}_p, \underline{y}_m) = f(\underline{E}_p, \underline{E}_m) \tag{4.22}$$

$$\underline{e} \equiv \underline{x}_p - \underline{x}_* = \underline{E}_p - \underline{E}_*, \tag{4.23}$$

so that the output error retains the physical meaning it needs to control the aircraft effectively using the adaptive algorithm, but the state error is defined properly so that the existing proof is still valid. It is unclear, however, how defining the state and output errors in different ways affects the physical meaning of the system and more work would need to be done to ensure that it wouldn't change the system entirely.

It may also be possible to modify the Lyapunov function for the system so that stability can be proven for the Euler rotation error defined state and output errors directly. Doing so would require taking a derivative of the quaternion error from Equation 3.7 and then manipulating it in such a way so that it fits into the existing MRAC stability proof. This is a daunting mathematical task, however, and this thesis was focused on the practical application of adaptive control rather than the mathematical theory, so only a small and unsuccessful attempt at going down this route was made.

It is also possible that redefining the Lyapunov function would lead to a new unique formulation of the adaptive control algorithm that behaves similarly to the MRAC method, but is technically a different control scheme all together. Potentially, this new control scheme could be much more effective at controlling a fixed-wing aircraft than MRAC and might eliminate some of the problems that were seen during simulations and flight testing. Investigating a new adaptive control formulation would be a very interesting topic for future works, but was out of the scope of this thesis.

4.2.2 MRAC Algorithm

The design of the Model Reference Adaptive Control algorithm will now be presented. If $(A_p + B_p G_e^* C_p, B_p, C_p)$ is strictly positive real (SPR) and the system of equations given in Equations 4.9, 4.10, 4.11, and 4.12 are totally consistent (i.e. they satisfy the matching conditions) with the adaptive control gain defined as

$$\dot{G}_e = -\underline{e}_y \underline{e}_y^T H \tag{4.24}$$

where $H \in \Re^{MxM} > 0$, and the control law for the system is defined as

$$\underline{u}_p = G_e \underline{e}_y \tag{4.25}$$

then the closed loop system is asymptotically stable with the output error of the system, $\underline{e}_y$, going to zero as $t \to \infty$. In practical terms for the transition-to-hover application, this means that the aircraft's attitude will asymptotically approach [0, 90, 0] degrees as $t \to \infty$.

H is referred to as the adaptive parameter matrix while G_e is the adaptive gain matrix. The values of the elements of H and the initial conditions for the elements of G_e are the only design decisions that have to be made to complete the adaptive controller. Practically, the adaptive controller works similarly to a simple proportional controller for each of three control axes, but the amount of gain in the system is

increased to help make up for lack of control authority if the output error is non-zero (i.e. the plant is not tracking the reference model perfectly). This property is what makes MRAC a potentially good fit for use as a transition-to-hover controller because low magnitude proportional gains are needed during high speed level flight while high magnitude gains are needed as the aircraft slows down and begins to hover.

To help make the adaptive controller easy to use as an addition to the ArduPlane firmware, the initial condition of G_{e_0} was chosen to be a diagonal matrix with the non-zero elements equal to the level flight proportional gains discussed in Section 4.1.2 such that

$$G_{e_0} = \begin{bmatrix} k_{p_{roll}} & 0 & 0 \\ 0 & k_{p_{pitch}} & 0 \\ 0 & 0 & k_{p_{yaw}} \end{bmatrix} = \begin{bmatrix} 0.15 & 0 & 0 \\ 0 & 0.27 & 0 \\ 0 & 0 & 0.35 \end{bmatrix}. \tag{4.26}$$

Setting the initial conditions this way essentially makes the adaptive controller mimic the performance of the flight tuned PID controllers when the adaptive controller is first initiated, but then the algorithm adds gain as needed to ensure proper tracking.

The values in H were chosen by trial and error through simulations and flight testing. The goal was to find values that were large enough so that G_e would grow quickly enough to prevent large output errors during the transition, but also small enough so that disturbances due to wind gusts would not cause large and unnecessary gain increases in relation to the success of the transition as a whole. Because of the $\underline{e}_y \underline{e}_y^T$ term in Equation 4.24, and the requirement that $H > 0$, there is no way for the magnitude of the diagonal values in G_e to decrease at any point while the adaptive controller is working. Thus, the design of the adaptive parameter matrix becomes very important in ensuring the control response does not become jittery and unpredictable

if the gains becomes too large. The final values of H were chosen to be

$$H = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}. \tag{4.27}$$

4.3 Reference Models

The HOVER_PID_REF and HOVER_ADAPTIVE flight modes both make use of reference models to smooth and lengthen the transition from 0 to 90 degrees pitch angle. Eight separate reference models were used to test the controllers' abilities to perform both fast and slow transition-to-hover maneuvers but all had the same general second order transfer function form of

$$\frac{\theta(s)}{u(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}. \tag{4.28}$$

where θ is the reference pitch angle, ω_n is the natural frequency in radians per second, and ζ is the damping ratio. As discussed in Chapter 4, the Command Generator Tracker produces a step function that is used as the input to the reference model for the HOVER_PID_REF and HOVER_ADAPTIVE flight modes which has the form

$$u(s) = \frac{90}{s} \tag{4.29}$$

so the transfer function for the step response of the reference model becomes

$$\theta(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2}\frac{90}{s} \tag{4.30}$$

which has the time domain solution

$$\theta(t) = 90 + (\theta_0 - 90)\mathrm{e}^{(-t\omega_n\zeta)}\left[\cos\left(t\omega_n\sqrt{1-\zeta^2}\right) - \frac{\sin\left(t\omega_n\sqrt{1-\zeta^2}\right)\left(\omega_n\zeta - \frac{(2\theta_0-180)\omega_n\zeta}{\theta_0-90}\right)}{\omega_n\sqrt{1-\zeta^2}}\right] \tag{4.31}$$

where θ_0 is the pitch of the aircraft at the time the transition maneuver is initiated. The rise time (t_r) and settling time (t_s) can be easily estimated for this type of second

order system by

$$t_r = \frac{1.8}{\omega_n} \tag{4.32}$$

$$t_s = \frac{4.6}{\zeta\omega_n} \tag{4.33}$$

which means that the reference model designer can easily chose the rise time by setting $\omega_n = 1.8/$(desired rise time in seconds). All eight reference models used in this thesis were designed with a damping ratio of 0.7 because that creates a step response with a maximum overshoot (M_p) of approximately 5%. Commanding the aircraft to go past 90 degrees pitch angle for a short period of time first helps ensure that the airplane will reach a fully vertical attitude, and then will also help stop any residual forward momentum the vehicle may have. Rise time values of 1, 2, 3, 5, 7, 10, 15 and 20 seconds were chosen to test both short and long duration maneuvers. Smaller rise time spacing was used for the faster transitions to help make comparisons between the flight modes that use the reference models and the HOVER_PID_STEP mode which tries to reach 90 degrees pitch angle as fast as possible. Step responses for the eight different reference models can be seen in Figure 4.5 followed by a summary of the approximate performance metrics in Table 4.4.

	Design Parameters		Estimated Performance Metrics		
Ref. Model No.	ω_n [rad/sec]	ζ [-]	t_r [sec]	t_s [sec]	M_p [%]
1	1.8	0.7	1.0	3.7	5.0
2	0.9	0.7	2.0	7.3	5.0
3	0.6	0.7	3.0	11.0	5.0
4	0.36	0.7	5.0	18.3	5.0
5	0.26	0.7	7.0	25.6	5.0
6	0.18	0.7	10.0	36.5	5.0
7	0.12	0.7	15.0	54.8	5.0
8	0.09	0.7	20.0	73.0	5.0

Table 4.4: Reference Model Design Parameters

It is important to note that only the pitch command was filtered through a second order reference model. The controllers try to hold zero roll angle and constant heading

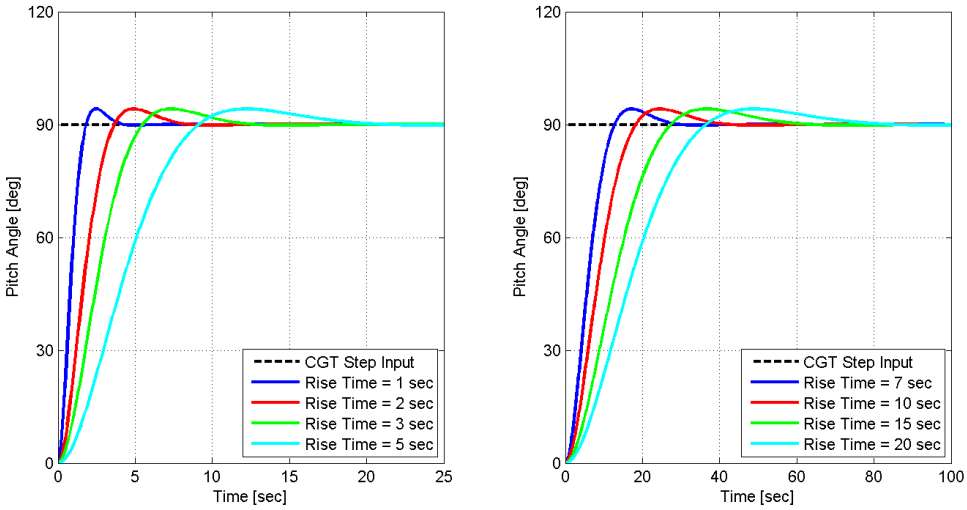


Figure 4.5: Pitch Angle Reference Models

before, during and after the transition so the resulting step input has a magnitude of zero throughout the maneuver, which means fitting reference models would be pointless.

4.4 Hover-to-Level Flight Control

The ability to transition the airplane from hover flight back to level flight is already included indirectly as part of the standard ArduPlane firmware. If the pilot switches the autopilot to STABILIZE, FLY_BY_WIRE or any other of the level flight modes while the aircraft is in the hover orientation, the attitude PID controllers (using the level flight gains shown in Table 4.1) will simply try to return the aircraft to 0 degrees pitch angle as soon as possible. This step input causes the aircraft to smoothly pitch the nose down, since there is limited elevator control authority when in hover, and slowly start to increase speed until steady level flight is achieved.

The throttle control method when transitioning back to level flight is dependent on the level flight mode that the autopilot was switched to. Fully autonomous flight

modes will control the throttle automatically using the feed forward method described in Section 4.1.3, whereas the throttle is controlled manually for pilot assisted modes like STABILIZE and FLY_BY_WIRE.

For the simulation and flight test results of the hover-to-level transitions discussed in Chapters 6 and 7, the autopilot was always switched from one of the three hover modes into the STABILIZE flight mode, and the throttle was set to a constant 75% throughout the maneuver. Setting the throttle to 75% ensured that the aircraft had enough power to accelerate to cruising speed quickly, and without losing too much altitude during the transition, so that the airplane was never in danger.

4.5 Control Logic Summary

A summary of the control logic and control gains used for each of the three new hover flight modes, and the STABILIZE mode used to transition the airplane back to level flight, is presented here. *Level* PID Gains refer to the attitude PID controller stabilization gains presented in Table 4.1, where as *Hover* PID Gains refer to the amplified control gains used to maintain stable hover that were presented in Table 4.2. Table 4.5 summarizes the attitude control logic for each of the different flight modes.

Attitude Control Logic Summary					
Flight Mode	Input	Pitch < 85 deg.		Pitch ≥ 85 deg.	
		Controller	Gains	Controller	Gains
HOVER_PID_STEP	Step	PID	Level	PID	Hover
HOVER_PID_REF	Ref. Model	PID	Level	PID	Hover
HOVER_ADAPTIVE	Ref. Model	MRAC	MRAC	PID	Hover
STABILIZE	Step	PID	Level	PID	Level

Table 4.5: Attitude Control Logic Summary

It is important to remember that when the plane is in HOVER_ADAPTIVE mode, the PID controller takes over control of the airplane from the adaptive controller when

a pitch angle of 85 degrees is reached. The hover PID controllers then maintain control of the aircraft for the remainder of the time the HOVER_ADAPTIVE flight mode is activated. The adaptive controller does not regain control of the airplane if the PID controllers let the pitch angle drop below 85 degrees.

Throttle Control Logic Summary		
PID Throttle Output	Pitch or Yaw Error > 5 deg.	Throttle Output
PID < 50%	yes	75%
	no	50%
50% ≤ PID ≤ 75%	yes	75%
	no	PID Output
PID > 75%	yes	PID Output
	no	PID Output

Table 4.6: Throttle Control Logic Summary

The throttle control logic is identical for all three hover flight modes and is summarized in Table 4.6. As a reminder, the throttle controller operates on the descent/climb rate error which is generated using a descent/climb rate command between ±6.5 feet per second which is manually set by the pilot. Additionally, the hover divergence logic controller is initiated when the magnitude of the pitch or yaw error exceeds 5 degrees. The throttle controller gains presented in Table 4.3 are the same for all three hover flight modes and do not change their values at any point during the flight. Finally, the throttle is controlled manually while the autopilot is in STABILIZE mode and was set to a constant 75% for the duration of the hover-to-level flight transitions.

5 SIMULINK SIMULATION

A Simulink simulation was constructed in order to help design each of the three new hover flight modes and establish the control logic that would later be implemented in the ArduPlane software. While many airplane simulators calculate the dynamics of the aircraft using stability derivatives, that method is dependent on the aircraft only experiencing small perturbations from a steady level flight condition. Since transition-to-hover maneuvers can push the aircraft into high angles of attack, and even deep stall regimes, the following simulator was designed using a combination of empirical methods and experimental data in an attempt to capture the effects of nonlinearities in post-stall aerodynamics.

The simulator uses Simulink's built in 6DoF (Quaternion) block, which is part of the Aerospace Blockset. The 6DoF block integrates the equations of motion using the aircraft's body axis forces and body axis moments as inputs. The forces and moments were calculated for each of the four major components of the aircraft (propulsion system, wing, horizontal tail, vertical tail, and fuselage) using custom Simulink blocks that are described in the following sections. The quaternion representation was chosen for integrating the equations of motion inisde the block, instead of Euler angles, in order to avoid the singularities that are caused in the Euler angle integration when the aircraft reaches ± 90 degrees pitch angle. More information on quaternions and their relationship to Euler angles can be found in Chapter 3.

Unless otherwise noted, x_e, y_e and z_e represent the position of the aircraft in the North-East-Down reference frame. Similarly, the body axis reference frame vectors are denoted by x, y and z where positive x is out the nose of the aircraft, positive y is out the right wing, and positive z is out the belly of the aircraft. Additionally,

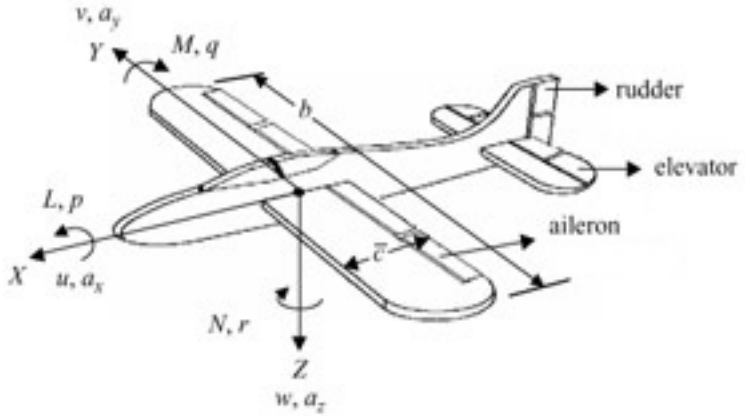


Figure 5.1: Body Axis Reference Frame

u, v and w are the body axis velocities, and a_x, a_y, a_z are the body axis accelerations which are aligned with x, y and z directions, respectively. The positive direction of the body axis rotation rates p, q and r as well as the body axis moments L, M, and N are defined by the right-hand-rule, as shown in Figure 5.1.

5.1 Wings and Stabilizers

The aerodynamic forces and moments generated by the wing and tail surfaces were all calculated using the numerical nonlinear lifting-line method outlined in [5]. This method was used to calculate the the lift distributions of the wing and tail surfaces after incorporating the effects of down-wash created by the wingtip vortices of three-dimensional wings. The lifting-line method is dependent on being able to calculate the circulation created by each two-dimensional airfoil of the discretized wing using the equation

$$\Gamma(y_k) = \frac{1}{2} V_\infty c_k (C_l)_k \tag{5.1}$$

where c_k is the chord length of the kth wing section and $(c_l)_k$ is the corresponding sectional lift coefficient. V_∞ is the local free-stream velocity at the nth wing section

which is calculated by

$$V_\infty = \sqrt{(u + u_{propwash})^2 + w^2} \tag{5.2}$$

for the wing and horizontal stabilizer, or

$$V_\infty = \sqrt{(u + u_{propwash})^2 + v^2} \tag{5.3}$$

for the vertical stabilizer, where $u_{propwash}$ is the propeller induced velocity. $u_{propwash}$ is equal to zero for the wing sections that are outside the propeller stream tube. Details on the calculation of the propeller induced velocity and the propeller stream tube diameter are given in Section 5.3. Since the sweep angles of the wings and stabilizers of the aerobatic airplane in this study are small, the effect of each cross-surface wind component was assumed to be negligible and was excluded from the V_∞ calculations.

The sectional lift coefficient is derived from wind tunnel data for a NACA 0012 airfoil for angles of attack between -180 and +180 degrees found in [10]. Sectional drag and moment coefficient values are also provided in [10] and all three coefficient values are found for the discretized wing sections by using the calculated local effective angle of attack as the reference value as shown in Figure 5.2. The advantage of using actual wind tunnel data for the sectional coefficients is that the nonlinear aspects of the high angle of attack aerodynamics are included when all of the discretized sectional forces and moments are integrated across the wingspan to get the total lift, drag, or moment for the wing, or tail surfaces. Similar data can be substituted for the NACA 0012 data if the simulated aircraft has different airfoils.

Empirical methods found in [1] and [29] were used to incorporate the effects of control surface deflections into the baseline aerodynamic coefficients from [10]. All control surface deflections (ailerons, elevator and rudder) were modeled as the deflection of a simple hinged flap. Empirical data from [29] was used to find the change in sectional lift coefficient due to flap deflection $\frac{\partial C_l}{\partial \delta_f}$, and the effectiveness parameter K_f, which are both dependent on the ratio of flap chord length to total airfoil chord

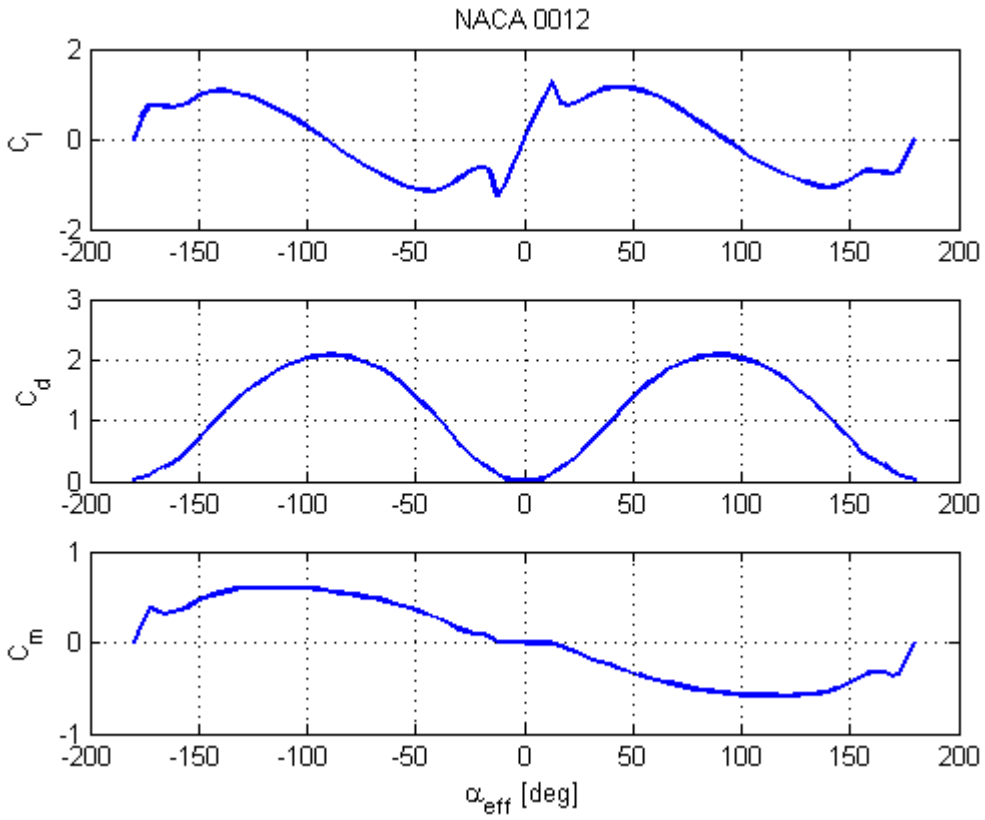


Figure 5.2: NACA 0012 Aerodynamic Coefficients

length, $\frac{c_f}{c}$. Empirical data from [1] was then used to find the change in sectional moment coefficient due to flap deflection $\frac{\partial C_m}{\partial \delta_f}$, which is also dependent on $\frac{c_f}{c}$. The change in sectional drag coefficient due to control surface deflection was assumed to be negligible and instead the increase in drag was accounted for in the induced drag term where

$$(\Delta C_{D_i})_{flap} = \Delta C_{L_{flap}} \sin \alpha_i \tag{5.4}$$

where α_i is the induced angle of attack and is related to the geometric angle of attack, α, and the effective angle of attack, α_{eff} that is calculated using the lifting line method in [5] by

$$\alpha_i = \alpha - \alpha_{eff}. \tag{5.5}$$

Additionally, the control surfaces were assumed only to have an effect on the aerody-

namics when the airfoil is not stalled (-15 deg $\leq \alpha_{eff} \leq +15$ deg). An example of the changes in sectional aerodynamic coefficients due to control surface deflections is shown in Figure 5.3.

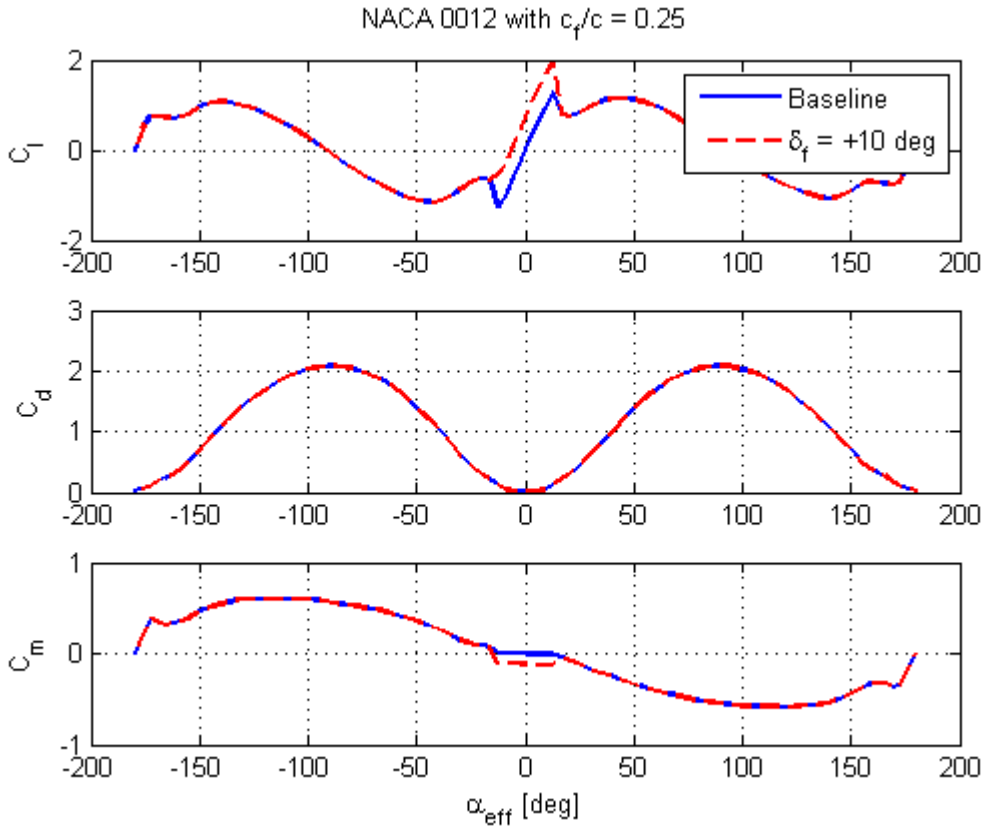


Figure 5.3: Aerodynamic coefficients for a NACA 0012 airfoil with $\frac{c_f}{c} = 0.25$ and $\delta_f = +10$ deg.

Once the effects of down-wash and control surface deflections were accounted for, the resulting aerodynamic coefficients were converted into sectional lift and drag forces and a sectional quarter-chord moment for each discretized wing section by

$$l_k = \frac{1}{2}\rho {V_\infty}^2 c_k (C_l)_k \tag{5.6}$$

$$d_k = \frac{1}{2}\rho {V_\infty}^2 c_k (C_d)_k \tag{5.7}$$

$$m_k = \frac{1}{2}\rho {V_\infty}^2 {c_k}^2 (C_{m_{sect}})_k \tag{5.8}$$

The lift and drag forces are then converted into sectional normal and axial forces that are aligned with the body axis by

$$f_{N_k} = l_k \cos \alpha_{avg} + d_k \sin \alpha_{avg} \tag{5.9}$$

$$f_{A_k} = -l_k \sin \alpha_{avg} + d_k \cos \alpha_{avg} \tag{5.10}$$

where α_{avg} is the average geometric angle of attack seen by the aerodynamic surface. The total normal and axial forces, as well as the moment that acts around the axis aligned with the span of the surface, were then determined by using the trapezoidal rule to integrate the sectional forces across the span of the surface by

$$F_N = \int_{-b/2}^{b/2} f_N dy = \frac{1}{2} \sum_{k=1}^{M} (y_{k+1} - y_k)(f_{N_{k+1}} + f_{N_k}) \tag{5.11}$$

$$F_A = \int_{-b/2}^{b/2} f_A dy = \frac{1}{2} \sum_{k=1}^{M} (y_{k+1} - y_k)(f_{A_{k+1}} + f_{A_k}) \tag{5.12}$$

$$M_{c/4} = \int_{-b/2}^{b/2} m dy = \frac{1}{2} \sum_{k=1}^{M} (y_{k+1} - y_k)(m_{k+1} + m_k) \tag{5.13}$$

where M is the total number of discretized wing sections and y_k is the distance from the body frame x axis to the location of the kth airfoil section. Forces like adverse yaw, which is a yawing moment caused by aileron deflection, were also calculated using the trapezoidal rule by summing the moment created by each sectional drag force around a specified axis. For example, the adverse yawing moment for the wing is calculated by

$$N_{wing} = \int_{-b/2}^{b/2} y_k f_{A_k} dy = \frac{1}{2} \sum_{k=1}^{M} (y_{k+1} - y_k) y_k (f_{A_{k+1}} + f_{A_k}) \tag{5.14}$$

Finally, the resulting forces and moments are then aligned with the body axis by

$$F_x = -F_A \tag{5.15}$$

$$F_y = 0 \tag{5.16}$$

$$F_z = -F_N \tag{5.17}$$

$$L = -\frac{1}{2}\sum_{k=1}^{M}(y_{k+1} - y_k)y_k(f_{N_{k+1}} + f_{N_k}) \tag{5.18}$$

$$M = F_N x_{CG} + M_{c/4} \tag{5.19}$$

$$N = \frac{1}{2}\sum_{k=1}^{M}(y_{k+1} - y_k)y_k(f_{A_{k+1}} + f_{A_k}) \tag{5.20}$$

for the wing and horizontal stabilizer, which lie in the XY plane, where x_{cg} is the distance from the vehicle center of gravity to the surface's aerodynamic center (positive value means aerodynamic center is closer to nose than vehicle CG), or

$$F_x = -F_A \tag{5.21}$$

$$F_y = -F_N \tag{5.22}$$

$$F_z = 0 \tag{5.23}$$

$$L = -\frac{1}{2}\sum_{k=1}^{M}(z_{k+1} - z_k)z_k(f_{N_{k+1}} + f_{N_k}) \tag{5.24}$$

$$M = \frac{1}{2}\sum_{k=1}^{M}(z_{k+1} - z_k)z_k(f_{A_{k+1}} + f_{A_k}) \tag{5.25}$$

$$N = -F_N x_{CG} - M_{c/4} \tag{5.26}$$

for the vertical stabilizer, which lies in the XZ plane.

5.2 Fuselage

In order to simplify the calculation of the aerodynamic forces created by the body of the airplane, the fuselage was modeled as the superposition of two low aspect ratio wings: one in the XY plane, and one in the XZ plane. Instead of using the lifting-line

method described in Section 5.1, baseline aerodynamic coefficients were determined based on geometric angle of attack using the NACA 0012 data in [10]. Helmbold's equation [18] was then used to apply a low-aspect ratio correction such that

$$C_L = k_{C_L} * C_{l_0} \tag{5.27}$$

where

$$k_{C_L} = \frac{1}{\sqrt{1 + (\frac{a_0}{\pi AR})^2} + \frac{a_0}{\pi AR}}, \tag{5.28}$$

C_L is the corrected 3-D wing lift coefficient, C_{l_0} is the baseline sectional C_l value found using [10], a_0 is the ideal lift curve slope 2π, and AR is the aspect ratio of the fuselage defined by

$$AR = \frac{\bar{D}_{fuse}}{l_{fuse}} \tag{5.29}$$

where l_{fuse} is the total length of the fuselage and $\bar{D}_{fuse}$ is the average diameter of the fuselage. The induced drag coefficient is then found by

$$C_{D_i} = \frac{{C_L}^2}{\pi e AR} \tag{5.30}$$

where

$$e = \frac{a_0}{(\frac{1}{k_{C_L}-1})\pi AR}. \tag{5.31}$$

The induced drag coefficient is then combined with the sectional drag coefficient to get the total drag coefficient such that

$$C_D = C_d + C_{D_i} \tag{5.32}$$

Finally, the sectional moment coefficient is assumed to be unaffected by the low aspect ratio of the fuselage so the value found using [10] based on the geometric angle of attack is used without any corrections being applied.

Once the three dimensional lift, drag and moment coefficients are determined for the fuselage in both the XY and XZ planes, the aerodynamic coefficients are then converted to normal and axial forces and then aligned with the vehicle's body axis using the same method outlined at the end of Section 5.1.

5.3 Propulsion System

Accurate modeling of the propulsion system for hovering fixed-wing aircraft is especially important since the propeller down-wash is the only airflow over the control surfaces while the airplane is hovering. Likewise, precise throttle control is needed to maintain constant altitude when hovering and it is important to capture the effects that propeller spin-up and wind-down have on the propeller forces and moments before being transfered to the airframe. It is preferable to use experimental data for determining RC propeller coefficients whenever possible since plastic RC sized propellers have the tendency to flex and twist when they are spinning, an effect that is very difficult to model using empirical methods like blade element and momentum theory. Modeling of the propulsion system was separated into two stages for each time step of the simulation. First, the 6 DoF forces and moments are calculated, and then the propeller stream tube solution is found since it is dependent on the force and moment results.

5.3.1 Propulsion Forces and Moments

The Yak 54 model used in this study has a single DC brushless electric motor. As described in [17], it is common to classify these types of motors based on three different parameters: the internal resistance R_i (Ω), the no-load current, I_0, and the Kv constant K_v (rpm/V). The torque created by the motor can then be found by

$$T_m = \frac{1}{K_v}\left(\frac{V_a - \frac{\omega}{K_v}}{R_i} - I_0\right) \tag{5.33}$$

where ω is the rotational speed of the motor in rad/sec, and V_a is the applied voltage.

The propellers used in this study are standard size plastic RC propellers that are classified by the diameter and pitch angle of the propeller. For example, a 9x4.7 propeller is a propeller with a 9 inch diameter and a pitch of 4.7 inches per revolution.

Instead of using blade element or momentum theory, experimental data from the online UIUC Propeller Database [4] was used to calculate the thrust produced by the propeller. An example of this data is shown in Figure 5.4. The advance ratio, J, is described in [9] as

$$J = \frac{V}{\eta * D} \tag{5.34}$$

where V is the flow velocity normal to the propeller disk, η is the rotation speed of the propeller in revolutions per second, and D is the diameter of the propeller. Additionally, the thrust and power produced by the propeller are related to the thrust and power coefficients by

$$C_T = \frac{T}{\rho \eta^2 D^4} \tag{5.35}$$

$$C_P = \frac{P}{\rho \eta^3 D^5} \tag{5.36}$$

and the power is related to the propeller torque by

$$P = 2\pi \eta Q_p. \tag{5.37}$$

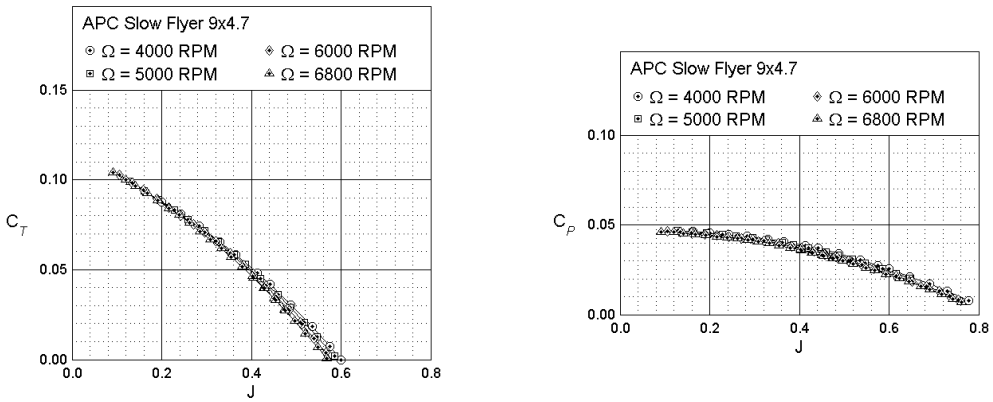


Figure 5.4: Example Propeller Coefficients

The rotational dynamics of the motor and propeller were modeled using the first order differential equation for simple rotational motion

$$J_{pm}\dot{\omega} = T_m - Q_p \tag{5.38}$$

where J_{pm} is the total combined axial moment of inertia of the propeller, motor rotator, and spinner assembly, and ω is the rotational speed of the propeller in radians per second. The Simulink model used to calculate the dynamic response of the motor assembly is shown in Figure 5.5.

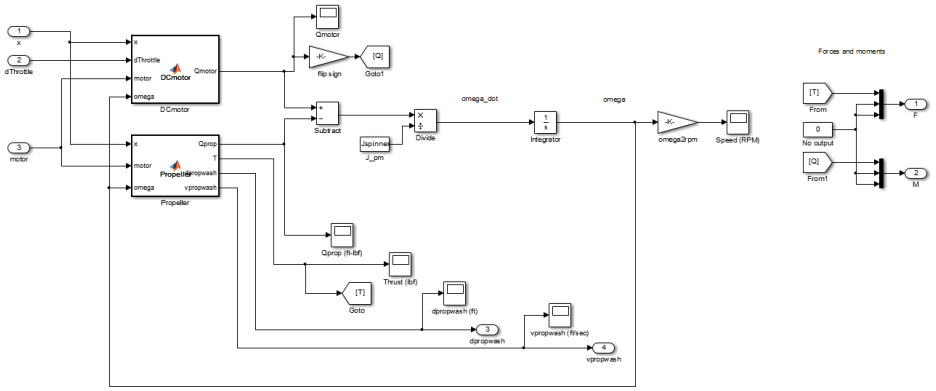


Figure 5.5: Simulink Motor Model

J_{pm} was calculated experimentally by attaching a constant diameter tube (with a known axial moment of inertia) to the motor assembly. A weight was then attached to one end of a string and the string was wrapped around the tube. The weight was then allowed to fall a measured distance, which unwound the string and accelerated the motor assembly. The time it took for the weight to complete the fall was then recorded. After the test was repeated numerous times and the average fall time was calculated, J_{pm} was calculated by

$$J_{pm} = \frac{m_w g D_{tube}^2 t^2}{8h} \tag{5.39}$$

where m_w is the mass of the weight at the end of the string, g is acceleration due to

gravity, D_{tube} is the diameter of the tube, t is the fall time in seconds, and h is the total distance the weight was allowed to fall.

Once the dynamic model of the motor and propeller was complete, it was easy to calculate the thrust produced by the propeller. At each time step, the motor speed and x-direction velocity of the aircraft was used to calculate the advance ratio of the propeller using Equation 5.34. The advance ratio was then used as a reference variable to determine the coefficient of thrust using the UIUC propeller database data [4] and then the actual thrust force was calculated using Equation 5.35. The resulting force was then applied in the positive x-direction of the aircraft's body frame. Since the torque produced by the motor was already calculated as part of the dynamic motor model, its value at each time step was simply pulled from the motor model and an equal and opposite moment was applied about the x-axis of the body frame. To complete the 6 DoF set of forces and moments, all forces in the y and z directions, as well as all moments in about the y and z axes, produced by the propeller and motor are assumed to be negligible.

5.3.2 Propeller Stream Tube

The last components of the propulsion system simulation are the velocity and diameter of the down-wash that is induced by the propeller on each of the aerodynamic surfaces of the aircraft. Equations found in [21] are first used to find the induced axial velocity at the propeller disk, $u_i(0)$, which is a function of the thrust produced by the propeller, T, such that

$$u_i(0) = \frac{-u_\infty + \sqrt{\frac{u_\infty^2 + 2T}{\rho \pi R^2}}}{2} \tag{5.40}$$

where u_∞ is the free-stream x-direction velocity of the aircraft, ρ is the free-stream air density, and R is the propeller radius. The ratio of the induced velocity at a specified location that is distance, l, downstream of the propeller disk, $u_i(l)$, to the induced

velocity at the propeller disk, $u_i(0)$, is then calculated by

$$K_{u_i} = \frac{u_i(l)}{u_i(0)} = 1 + \frac{l/R}{\sqrt{1 + (l/R)^2}}. \tag{5.41}$$

K_{u_i} is calculated for each of the aerodynamic surfaces on the aircraft (wing, fuselage, horizontal and vertical stabilizers) by setting l equal to the distance from the propeller disk to the aerodynamic center of the respective surfaces. The actual propeller induced velocity for each surface is then calculated by

$$u_i(l) = K_{u_i} u_i(0) \tag{5.42}$$

and it is important to remember that the induced velocity for each surface must be added to the free-stream velocity to get the actual air velocity over the respective surface.

Finally, the diameter of the propwash at distance l from the propeller disk is calculated by

$$D_{pw}(l) = 2\sqrt{\frac{(\pi R^2)(u_\infty + u_i(0))}{\pi(u_\infty + u_i(l))}}. \tag{5.43}$$

The induced velocity at each location l is assumed to be uniform throughout the diameter of the stream tube. It is also important to note that the variation in induced velocity along the chord length of the aerodynamic surfaces is ignored and the induced velocity at the aerodynamic center of each surface is used for the lift calculations in Section 5.1.

6 SIMULATION RESULTS

The primary purpose of the simulator discussed in Chapter 5 was to help design the control logic and establish initial controller gains for the hover flight modes before ever actually attempting an autonomous transition-to-hover maneuver in a real aircraft. After the initial simulation based designs were complete and the first flight tests were attempted, a combination of flight test base tuning and simulation based tuning was used to determine the final control logic and gains that are presented in Chapter 4. In order to get a sense of how accurate the simulator was compared to actual flights, and to get an idea of what to expect during flight tests, simulations were run with the intent of mimicking the 68 different conditions that were planned to be tested in flight.

Each of the three hover flight modes were simulated under different conditions to see how well they handled different types of transitions. For all three of the hover flight modes, each combination of controller type and reference model design was simulated at four different approach speeds: 40, 60, 80, and 100 ft/sec. The result was 4 different conditions for the HOVER_PID_STEP mode, and 32 different conditions for both the HOVER_PID_REF and HOVER_ADAPTIVE flight modes. Each controller, reference model, and approach speed combination was then simulated 5 times, with varying wind speeds, in order to get a measure of the repeatability of the results for a total of 340 separate simulations. The wind speed was set to 0, 5, 10, 15, and 20 feet per second for each of the five simulations respectively, to examine how well the controllers perform in different wind conditions. Simulink's built in Dryden Turbulence Model block was used to implement the simulated wind so that the wind disturbances had both rotational and lateral variability and simulated actual

wind conditions more closely. 20 feet per second was chosen as the maximum value because it is the approximate wind speed where manually controlled RC flights start to become hazardous.

Each simulation was initiated with the aircraft pointing due North, with nose and wings level to the horizon (i.e. 0 degrees roll, pitch and yaw), at an altitude of 100 feet, and at the XY origin of the arbitrary Earth reference frame. Downrange distance traveled was used as a basis for comparison among the different controllers and is defined as the distance the aircraft travels only in the Earth's x-direction. Movement in the y-direction is ignored to eliminate the influence of lateral wind drift and produce comparable plots to the flight test results. The altitude change was also used to compare controller performance and was always calculated in reference to the altitude at the beginning of the maneuver, instead of the ground, to make comparisons between simulation and flight test results easier. Additionally, ground speed is measured as the magnitude (independent of direction) of the velocity of the aircraft's center of mass in the XY plane (vertical velocity is ignored) to mimic the GPS based ground speed measurement taken during flight tests.

Throttle controller input was always set to 0 ft/sec descent/climb rate for the entire duration of the simulations. This was done so that any variations in altitude are caused by either the bleeding off of excess kinetic energy, or from the hover divergence logic controller needing to increase throttle in order to maintain stable hover, and not from manually changing the desired descent/climb rate.

The success or failure of each simulation was dependent on whether the aircraft was able to transition to, and then maintain stable hover, for at least 15 seconds (measured from the time the transition is initiated), or for a time equal to the reference model rise time multiplied by five, whichever is greater. If the magnitude of the pitch or yaw error ever exceeded 45 degrees for more than one second during the simulation,

the aircraft was determined to have diverged from stable hover and the simulation was stopped. The one second time delay before stopping the simulation was used to mimic a pilot observing the aircraft and waiting to see if the attitude stabilization could recover on its own before disengaging the autopilot and manually returning the airplane back to a safe orientation.

Finally, a complete record of the data collected from each of the 340 simulations is available in Appendix A.

6.1 PID Control Step Response

Simulations of the HOVER_PID_STEP fight mode showed that the aircraft was able to transition to and maintain stable hover for 15 seconds for each of the 20 different simulations that were run for this flight mode. As one would expect, the transition looked like a simple pull-up maneuver where a pilot would simply start in steady level flight, pull back hard on the elevator to increase pitch to 90 degrees as fast as possible, and then let gravity bleed off the residual kinetic energy until the aircraft stopped climbing. An example of the plotted results from one of these simulations is shown in Figure 6.1. 60 ft/sec was used as the example approach speed because it is the approximate cruise speed when controlling the aircraft manually, and 10 ft/sec was chosen for the wind speed because it is the intermediate value.

When looking at Figure 6.1 a few interesting details stand out. First, it appears that the PID controllers have a more difficult time controlling the roll angle of the aircraft compared to the pitch and yaw angles. This makes sense since most of the aileron surface area is outside of the propeller stream tube so the ailerons have very little control effectiveness when in hover. This means that any roll disturbances caused by motor torque or wind take a longer time for the ailerons to correct.

The second interesting thing in Figure 6.1 is that the pitch angle seems to undergo

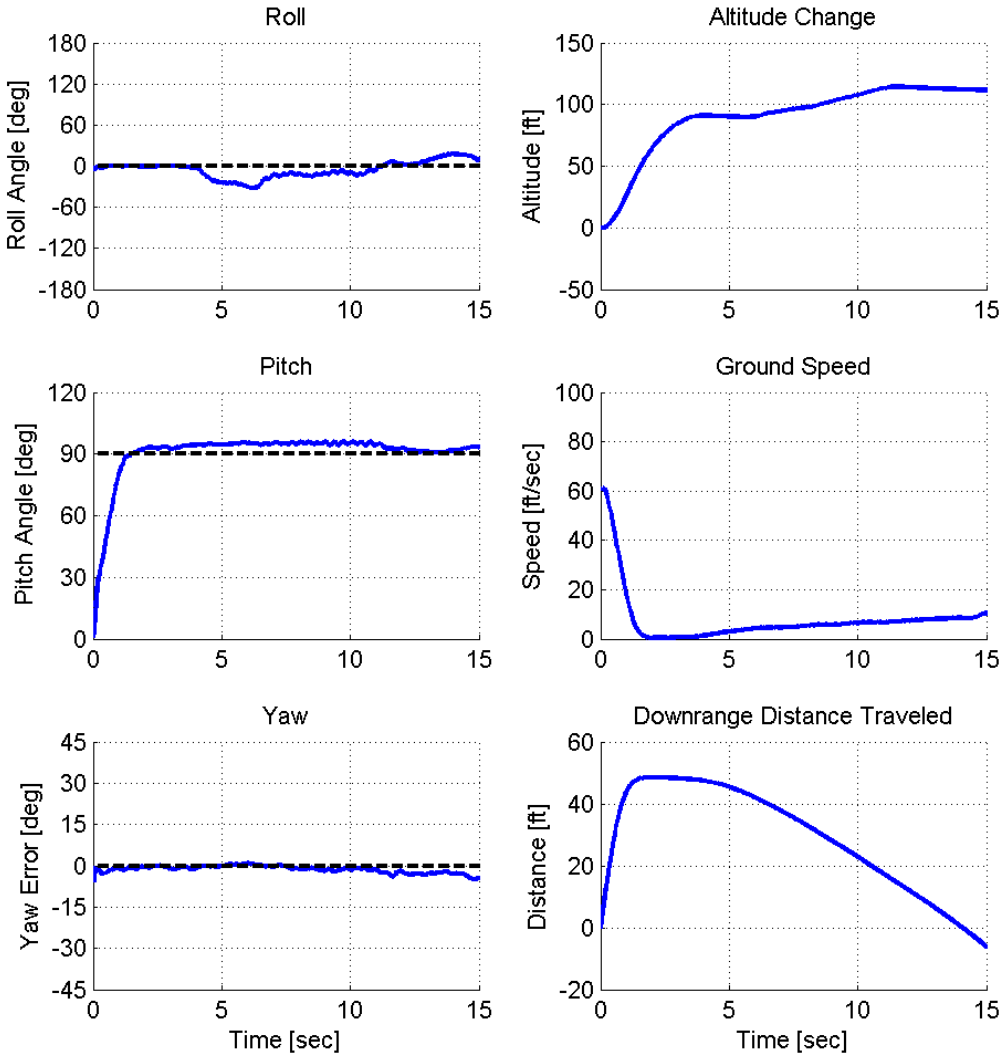


Figure 6.1: PID Control Step Response Simulation Example: HOVER_PID_STEP simulation results with approach speed = 60 ft/sec, and wind speed = 10 ft/sec.

very small oscillations between the 7 and 11 second marks. These are caused by the hover divergence logic controller going on and off as the pitch angle oscillates around 95 degrees. As soon as the pitch error exceeds 5 degrees the throttle is increased to 75%. This quickly increases the air velocity over the elevator, which in turn increases control effectiveness and decreases the magnitude of the pitch error. However, as

soon as the pitch error is back under 5 degrees, control of the throttle is returned to the descent/climb rate PID controller, the extra control effectiveness is lost, and the pitch error grows again. This process is repeated several times until the integral term in the elevator PID controller finally builds up enough magnitude to eliminate the steady state pitch error about 12 seconds into the simulation.

These pitch angle oscillations have a very interesting effect on the aircraft's altitude. For the first 5 seconds of the simulation the aircraft's altitude changes the way one would expect for a pull-up style maneuver. The altitude first changes gradually as the aircraft starts to pitch up and then increases rapidly once the nose is pointed straight up until all of the kinetic energy dissipates and the altitude PID controller takes over to maintain constant altitude. Because the pitch oscillations cause the throttle to be increased to 75% to aid control effectiveness, however, the airplane gradually starts to climb due to the excess thrust. Once the pitch oscillations stop at about 11 seconds, throttle control is handed back over to the altitude PID controller which is then able to hold altitude nearly constant for the remainder of the simulation. These short periods of altitude increase were seen throughout both the simulations and flight tests and were usually caused by the autopilot needing to increase the throttle in order to maintain attitude control in the presence of wind disturbances.

Probably the most interesting portion of the results in Figure 6.1 is how much the aircraft's distance traveled varied throughout the simulation. Since only the aircraft's attitude and altitude are controlled directly, the airplane is allowed to drift laterally as much as it needs while trying to maintain stable hover. In Figure 6.1, the simulated plane first travels about 48 feet downrange during the transition but then actually drifts all the way back over the starting position as the flight goes on. This backwards drifting is caused by a combination of the wind and the pitch oscillations. All simulations and flight tests were done into a head wind in order to help increase

control surface effectiveness and to help keep the airplane closer to the pilot for safety. The ground speed plot shows how the magnitude of the aircraft's ground speed first drops to 0 ft/sec at the end of the transition, and then slowly increases to about 10 ft/sec by the end of the simulation. The period of time where the pitch is greater than 90 degrees also meant that the thrust vector was pointed back towards the starting position slightly, instead of straight up in the air, which worked with the wind to build momentum back towards the starting position. This thrust vector induced drifting also caused the aircraft to drift side to side during some of the tests when there was large enough yaw angle error but this type of drifting was not used as a basis for comparison between any of the tests.

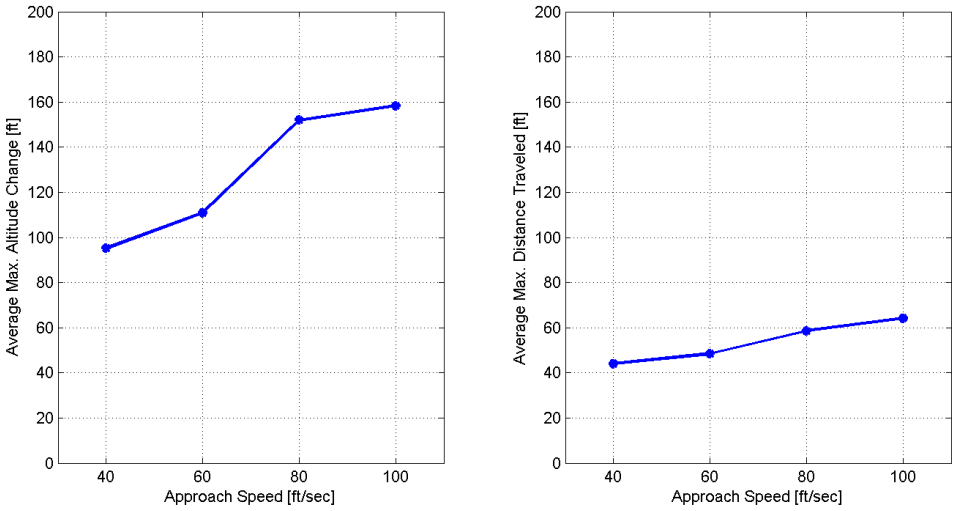


Figure 6.2: PID Control Step Response Simulation Results: Average altitude change and average distance traveled both increase as a function of the aircraft's approach speed.

Figure 6.2 shows how the average altitude change and average distance traveled of the aircraft varied based on the approach speed for the HOVER_PID_STEP simulations. As one would expect, both the altitude change and distance traveled increase as a function of approach speed. Since kinetic energy is a function of velocity squared it is no surprise that the altitude change looks like a quadratic between 40 and 80

ft/sec, since all of the initial kinetic energy has to be converted to potential energy before the aircraft stops climbing. It is strange, however that the altitude change does not keep increasing quadratically when the approach speed is increased from 80 to 100 ft/sec. This is most likely due to nonlinear aerodynamic effects that happen at the beginning of the transition maneuver. When the approach speed is less than or equal to 80 ft/sec the airplane is likely traveling slow enough that large amounts of flow separation does not occur on the wings as the pitch angle is increasing from 0 to 90 degrees. At 100 ft/sec, however, the elevator has enough control authority to rotate the aircraft to a very high angle of attack while the aircraft still has large amounts of forward momentum. This likely causes the wing to stall and the flow to separate which dissipates much more kinetic energy compared to a fully attached flow, and as a result means less altitude gain before the airplane stops climbing.

While the altitude change nearly doubles by increasing the approach speed from 40 to 100 ft/sec, the distance traveled only increases by about 50%. This is likely due to he fact that the pitch angle step response is analogous to the airplane making a minimum radius turn. Since the minimum turn radius of an aircraft is directly related to its maximum lift coefficient, which does not change very much with the relatively small speed changes, it is not surprising that the distance traveled was not affected by the speed increase as much as the altitude change was.

Since all twenty of the HOVER_PID_STEP simulations were successful, the only way to examine the effect increasing wind speeds had on the results was to examine how the altitude and distance results changed. Figure 6.3 shows how the altitude and distance traveled changed as the simulated wind speed was increased for each of the four different approach speeds. Neither the maximum altitude change or the maximum distance traveled are affected by the change in wind speed much, except for a few small jumps in the altitude results. What is even more interesting is that the few altitude jumps that were seen are not consistently increasing or consistently

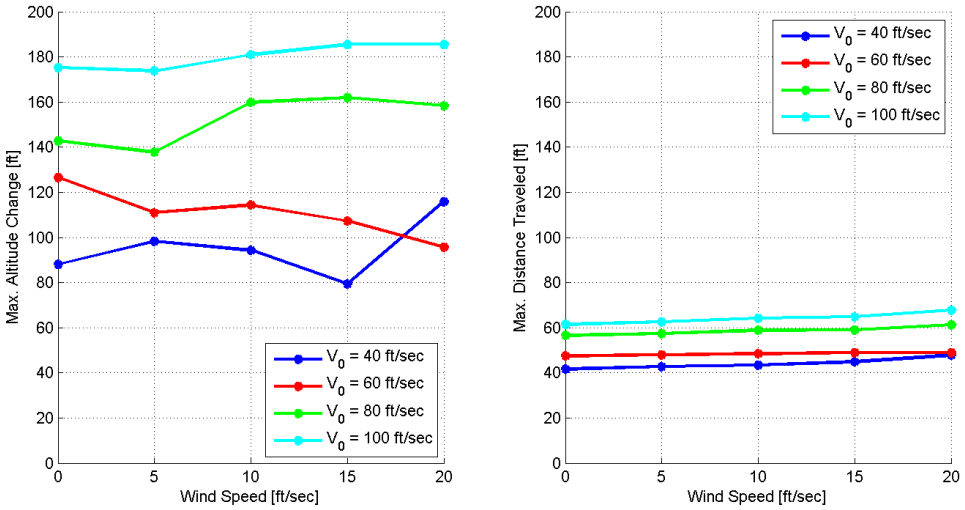


Figure 6.3: PID Control Step Response Simulation Wind Sensitivity

decreasing, but rather a combination of the two suggesting there is no correlation at all between simulated wind speed and the altitude change during the transitions. The jumps were more likely caused by the fact that the wind disturbances are introduced using Simulink's Dryden Turbulence Model block which introduces random linear and rotational air speed disturbances that are uniformly centered around the user specified wind speed and wind direction. The increases in altitude change when starting at the same approach speed are likely produced by pure luck when a random wind gust happened to occur at just the right time in order to force the pitch or yaw error to grow and force the hover divergence throttle to engage. This type of occurrence was seen many times during actual flight testing so it was encouraging to see that it showed up in simulation as well.

Overall, the HOVER_PID_STEP simulations showed that a PID controlled step response was a robust and fairly easy to implement way of completing a fixed-wing transition-to-hover maneuver. The main concern with any aircraft simulation, though, is that the simulation often has a difficult time capturing all of the nonlinear aerodynamic effects seen by a real aircraft, especially when operating at high angle

of attack. Flight tests are needed to show whether the real aircraft can maintain controllability while at high angle of attack and whether a simple step response is too violent of a maneuver to be used repeatedly in real flight. Observability problems may also become a factor. While time delays were added to the simulation signals to mimic the autopilot sampling data at 50 Hz, no variability was introduced to simulate the measurement errors of the sensors. The sensors used to calculate the aircraft's attitude in flight may not be able to keep up with the rapid angular accelerations seen during the transitions, which would cause the PID controllers to operated on Euler angle values with large errors in them, and possibly change the effectiveness of the HOVER_PID_STEP flight mode.

6.2 PID Control Reference Model Response

By changing the PID controller input from a step function to a second order reference model, the HOVER_PID_REF flight mode was intended to produce similar results to the HOVER_PID_STEP mode but also give the user the option of lengthening the transition time by changing the reference model design. Simulations of the PID control reference model response were set up the same way as the step response simulations. The only difference was that each test was conducted with the desired attitude being determined by one of the eight reference models discussed in Section 4.3. Figure 6.4 shows how the transition-to-hover maneuver changes by using a one second rise time reference model compared to the step response shown in Figure 6.1. Throughout this section, the desired reference model is shown with a dashed line and the measured system response is showed with a solid line of the same color.

Interestingly, the step response and one second rise time reference model response produce nearly identical results with respect to the roll, pitch and yaw errors. The pitch PID controller struggles to track the reference model (shown in dashed red)

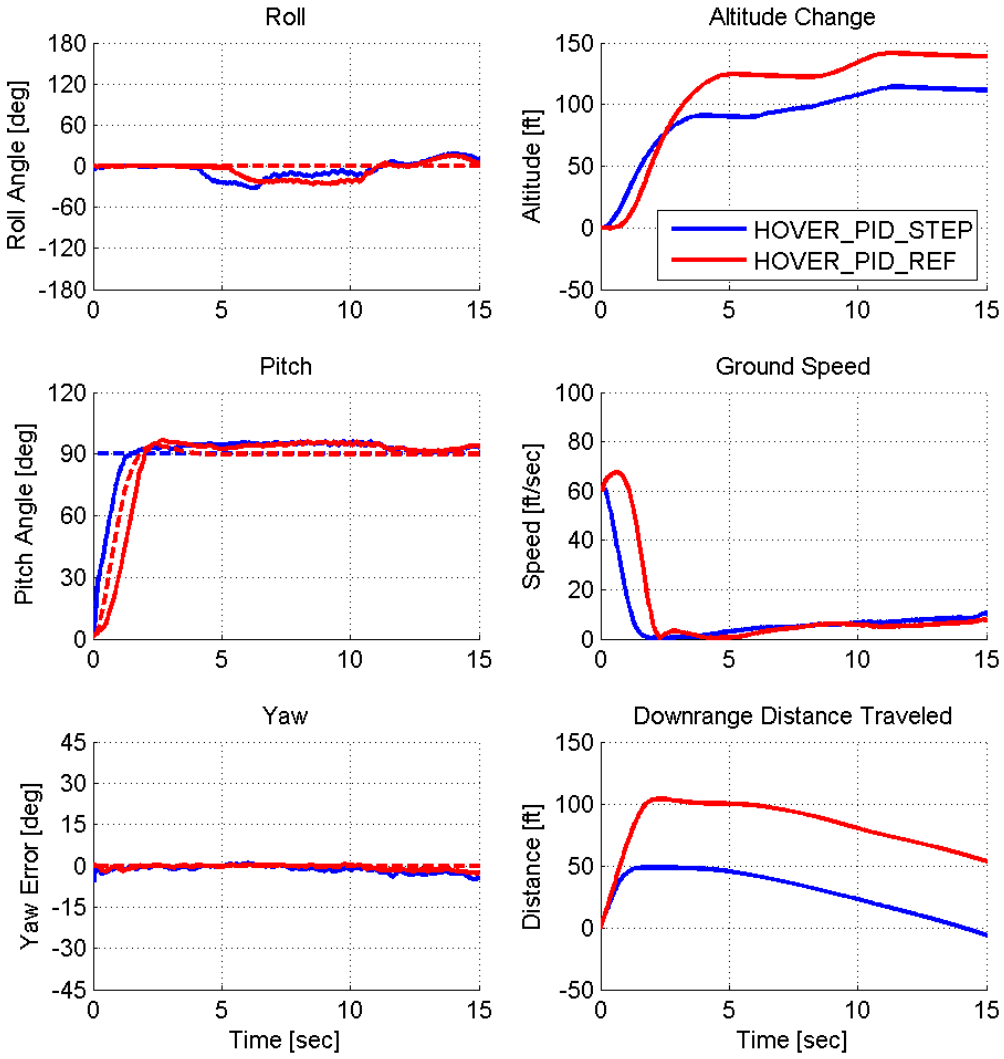


Figure 6.4: PID Control Reference Model Response Simulation Example: HOVER_PID_REF simulation results (rise time = 1 sec, approach speed = 60 ft/sec, wind speed = 10 ft/sec) compared to analogous step response.

perfectly during the transition, but then experiences he same pitch oscillations as he step response before finally eliminating the steady state error.

The most interesting differences between the two transition methods are in the altitude change, ground speed and distance plots. The reference model response shows that the airplane climbs an additional 30 feet during the initial transition before

finally stopping its climb and starting its hover. The ground speed plot shows that the aircraft actually speeds up to almost 70 ft/sec during the first couple seconds of the simulation which causes the distance traveled to increase and also increases the initial altitude change during transition. The speed increase is due to the aircraft throttling up to attempt to maintain constant altitude while at low pitch angle. Once the desired pitch angle starts to increase and the plane starts to pull up, the controller throttles back to the minimum 50% and lets the airplane coast upwards until it stops climbing.

Even though the initial altitude change is larger, the reference model response has a longer period of time where the aircraft is hovering at constant altitude when compared to the step response. The reference model response is able to hold altitude constant between the 5 and 8 second marks where as the step response is only able to hold constant altitude between the 5 and 6 second marks. This is because the slightly smoother reference model based transition creates a more stable starting point for the actual hovering and allows more time to pass before wind disturbances have enough effect for the hover divergence throttle to be needed. The more stable starting point also means that the divergence throttle increase is needed for a shorter period of time to help reject the wind disturbances compared to the step response.

Figure 6.5 shows how increasing the reference model rise time from 1 second to 2 or 3 seconds has a similar effect as switching from a step response to the 1 second reference model did. The longer the rise time is the more time the aircraft spends at a low pitch angles where more speed is needed to keep altitude constant. The longer periods of nearly level flight cause the maximum downrange distance traveled to increase dramatically but also cause the maximum altitude change to increase incrementally due to the extra initial kinetic energy.

Figure 6.6 shows that if the approach speed is increased to 80 ft/sec for the same

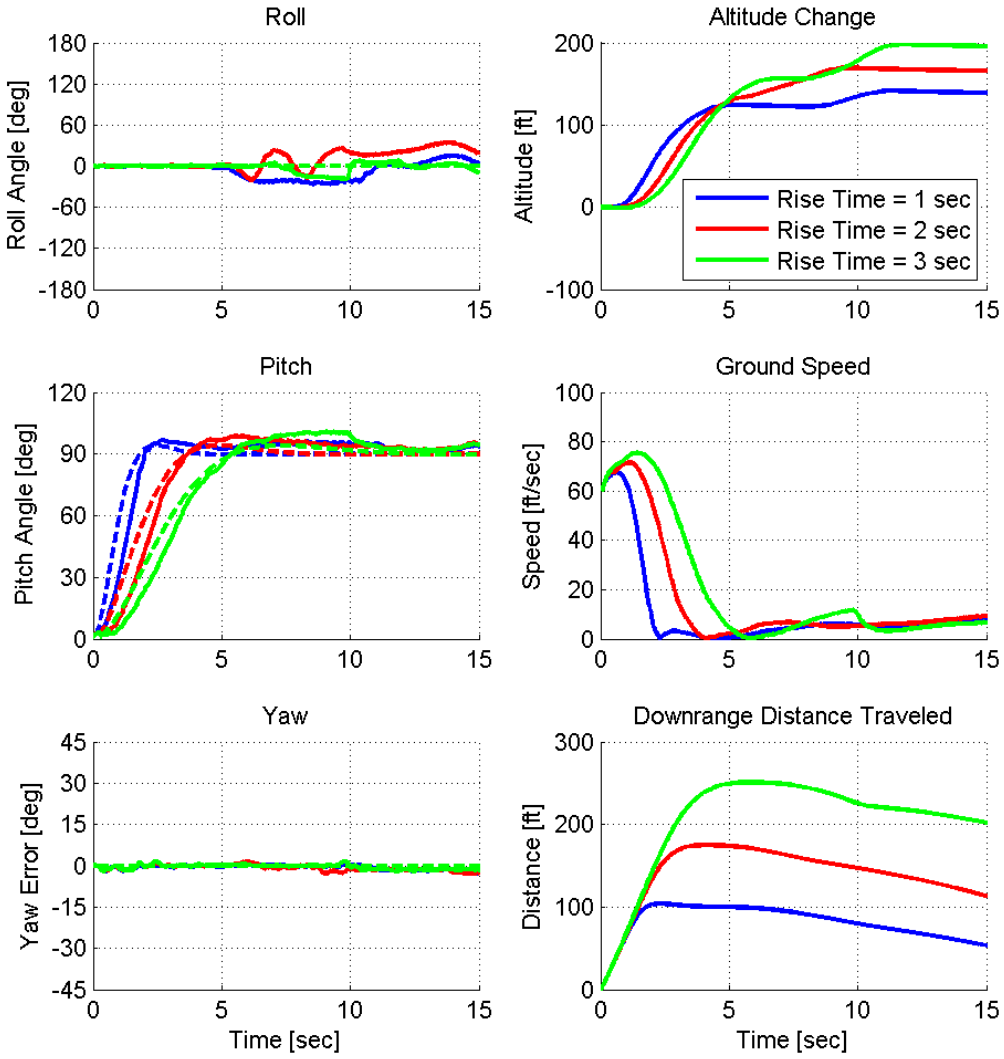


Figure 6.5: PID Control Simulation Variation With Reference Model Rise Time

three reference model responses, the aircraft no longer experiences an initial speed increase to try and hold altitude constant. While the maximum distance traveled still increases drastically when the rise time is increased because of the extra time that is spent at low pitch angles, there is less difference between the maximum altitude change for the three simulations since the airplane never accelerates at the beginning of the simulation. The differences in altitude change in Figure 6.6 are simply due to

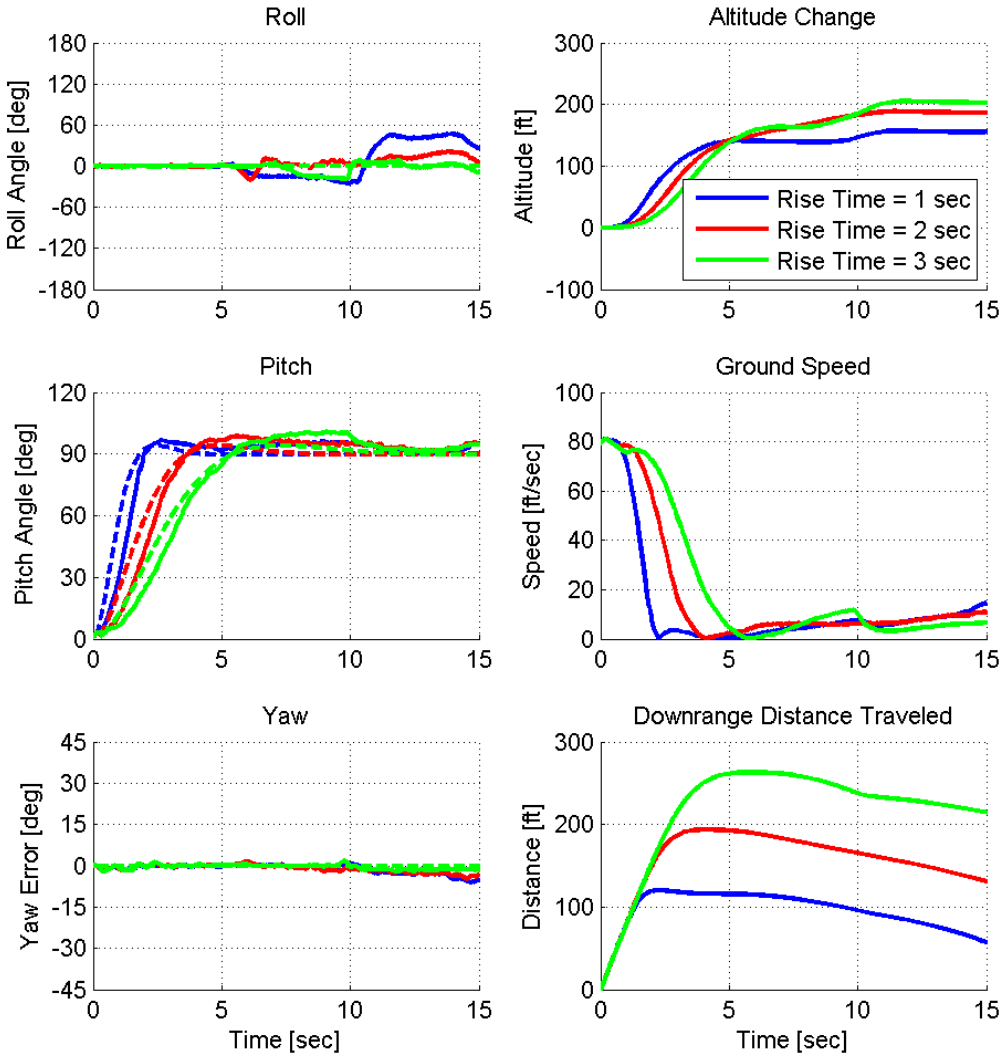


Figure 6.6: PID Control Reference Model Response Simulation Variation With Approach Speed

the need for the hover divergence throttle to engage and help keep the pitch and yaw errors within ±5 degrees at different points in the simulation.

Figure 6.7 shows that the PID controllers start to have trouble maintaining stable hover when the reference model rise time is increased to 5 seconds. As the reference model rise time is increased, the aircraft spends more and more time at intermediate

pitch angles where it is difficult to maintain control. This likely causes a large amount of integrator windup in the pitch axis PID controller which then causes a large overshoot in the pitch response once the 85 degree mark is hit and the gain magnitudes are increased. This overshoot in the pitch axis can be seen in Figure 6.7 at the 15 second mark, where the pitch angle peaks at about 115 degrees. The pitch axis PID controller then takes nearly the entire rest of the simulation to eliminate the steady state error and return back to a perfectly vertical orientation.

Interestingly, the large pitch error does not cause a large altitude increase like the ones seen in previous simulation plots. This is probably a result of the thrust vector being so far away from vertical that the aircraft was only able to maintain the current altitude even at full throttle. This theory is supported by the large increase in ground speed seen at the 15 second mark which meant that the aircraft started drifting rapidly due to the non-vertical thrust vector. It was just a coincidence that the 75% divergence throttle setting produced just enough thrust to maintain constant altitude while the aircraft had a pitch error of about 10 degrees between the 17 and 22 second marks.

Once the reference model rise time is increased passed 5 seconds, the PID controllers struggle to successfully complete the hover simulation as shown in Figure 6.8. The simulation stoppages in all four cases are caused by the yaw error exceeding 45 degrees for more than one second. When the aircraft gets close to the end of its transition and the airflow over the rudder is very low for an extended period of time, wind disturbances have the opportunity to push the tail one way or the other. When the gravity vector that originates at the airplane's center of mass is no longer parallel to the centerline of the airplane, the moment created by the gravity force continued to turn the nose of the aircraft back towards the ground. With the minimal airflow, the rudder does not have enough control authority to counteract the gravity moment and the aircraft diverges.

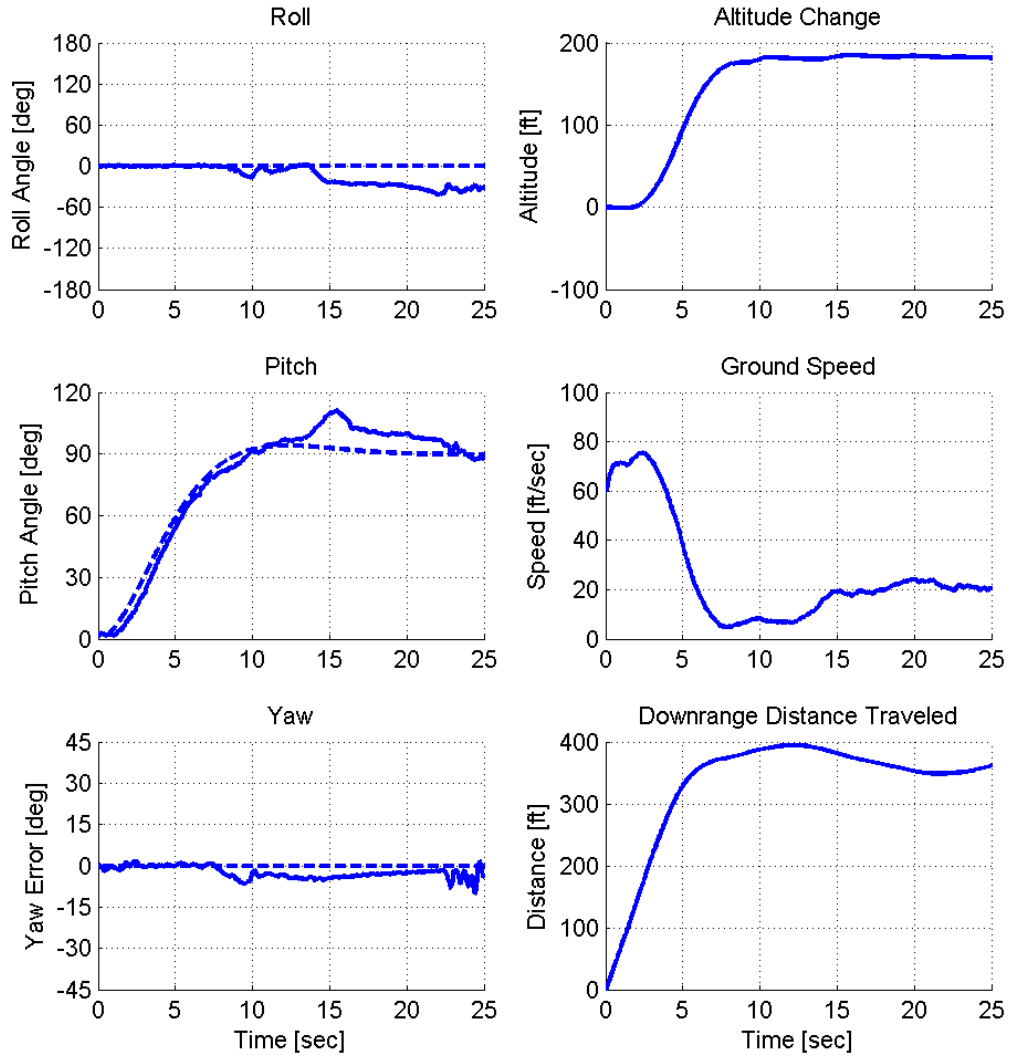


Figure 6.7: PID Control Reference Model Response Simulation Struggles At Rise Time = 5 sec

This type of divergence was by far the most common way for the HOVER_PID_REF flight mode to fail during both simulations and actual flight tests, so the simulations were useful for knowing what to look for during flight tests. However, the yaw divergence failures happened much more frequently during simulations than they did during actual flight test. The increased level of success of the HOVER_PID_REF flight mode during flight tests is discussed in Chapter 7, but this discrepancy be-

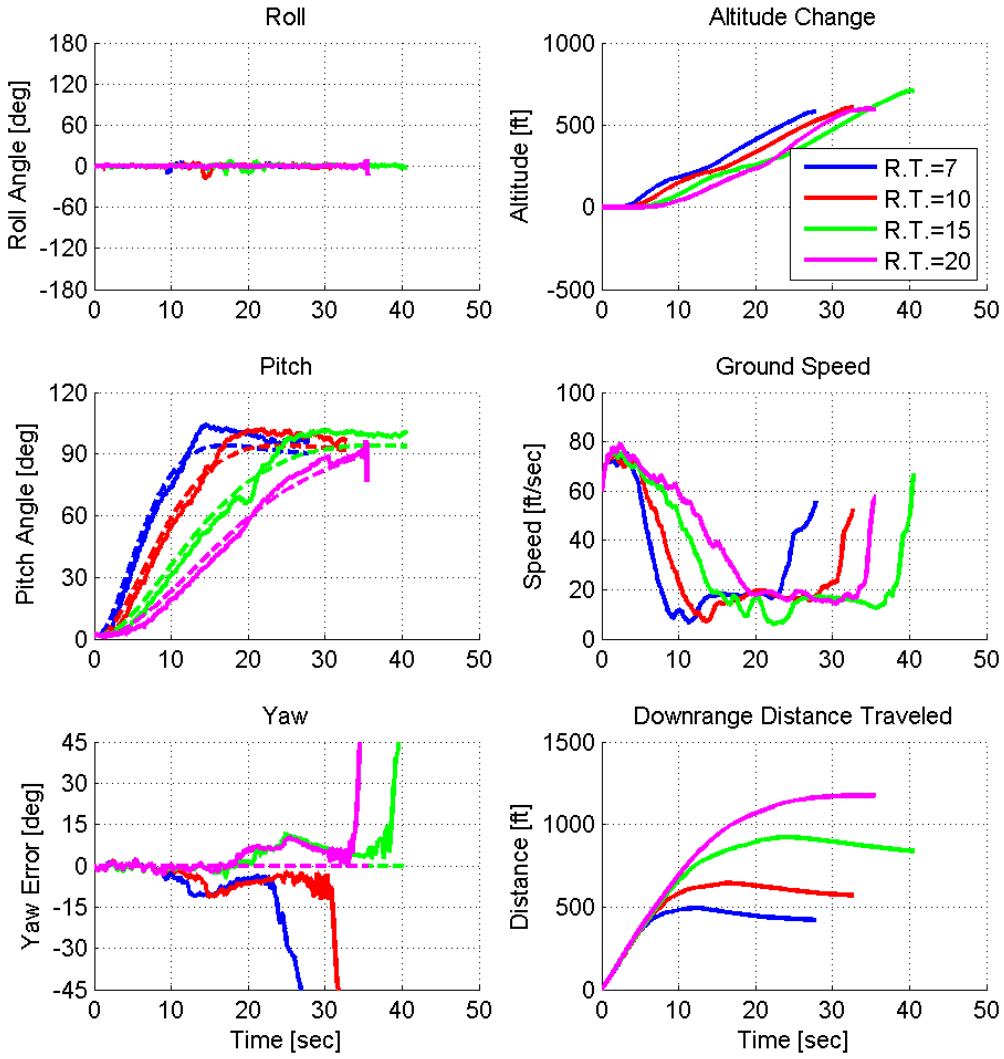


Figure 6.8: PID Control Reference Model Response Simulation Failures

tween simulation and flight test results was likely due to the difficulty involved with simulating the effects of the large aerobalance that is at the top of the actual Yak 54 model's rudder.

The actual aircraft was able to counteract the gravity moment created by yaw errors at high pitch angles more regularly than the simulated aircraft was, so the simulation failures were likely caused by inaccuracies in the simulated airframe as

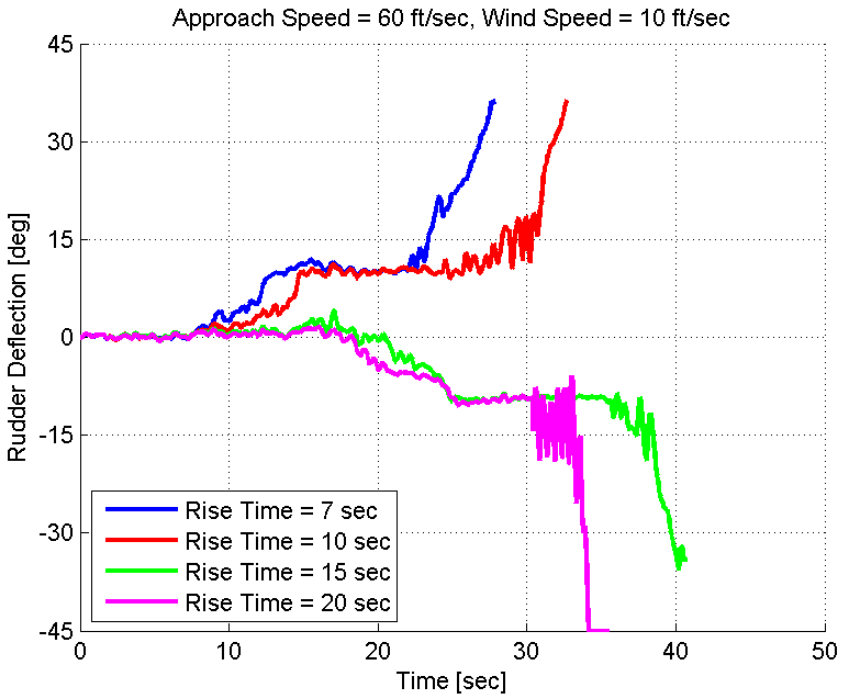


Figure 6.9: PID Control Reference Model Response Simulation Failures: Control Surface Deflections

opposed to the design of the rudder PID controller. This is demonstrated in Figure 6.9, which shows the control surface deflections for the results in Figure 6.8, where the rudder deflections do grow rapidly, and in one case even hits the maximum deflection of ± 45 degrees, as the yaw error starts to increase. Increasing the gains of the rudder PID controller may help delay this type of failure, but it appears that once the simulated aircraft starts to lean too far to one side there is no way to maintain stable hover. The simulated rudder forces simply do not produce a large enough moment to counteract gravity and correct the yaw error, and changes would need to be made to the simulator in order to generate more realistic results.

The probability of successful transition and hover for the HOVER_PID_REF simulations is shown in relation to both the reference model rise time and the approach speed in Table 6.1. This table shows that the probability of successful transition and stable hover shares an inverse relationship with rise time of the user specified

PID Control Reference Model Response Simulations: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
5	80%	60%	80%	40%	65%
7	40%	20%	20%	20%	35%
10	20%	0%	40%	0%	15%
15	40%	0%	0%	20%	15%
20	0%	0%	0%	0%	0%
Cumulative	60%	48%	55%	47%	53%

Table 6.1: PID Control Reference Model Response Simulations: Probability of Success

reference model. As discussed previously, all of the tests performed with a rise time of three seconds or less were successful but the probability of success decreases very quickly once the rise time is increased past five seconds. Again, nearly all of the HOVER_PID_REF simulation failures were due to the yaw divergence problem seen in Figure 6.8, more examples of which can be seen in Appendix A. The longer duration transitions simply allow for more time where a disturbance can push the tail of the aircraft out of alignment while the aircraft is at a high pitch angle and a low speed, at which point the simulated rudder does not have enough control authority to correct the yaw error.

Figure 6.10 shows the average maximum altitude change and average maximum distance traveled by the aircraft for each of the different HOVER_PID_REF approach speed and rise time combinations. The results of the HOVER_PID_REF simulations were just like the HOVER_PID_STEP results in that they were not sensitive to changes in simulated wind speed. Because of this, the results of five repeated tests for each approach speed and rise time combination were averaged to produce the plots in Figure 6.10. The different approach speeds were plotted as separate lines however

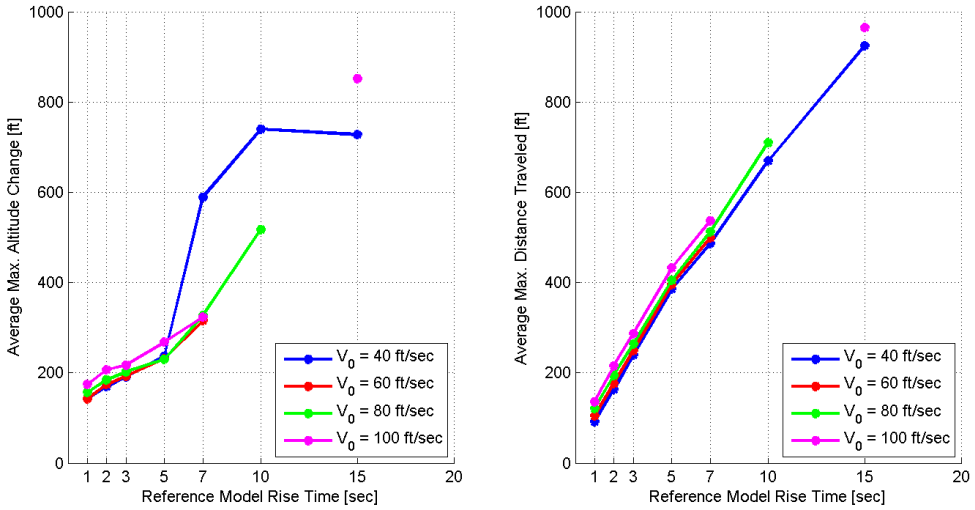


Figure 6.10: PID Control Reference Model Response Simulation Results: Simulation results show that altitude change and distance traveled are both fairly predictable for short rise times, regardless of approach speed (V_0).

since Figure 6.2 showed that altitude change and distance traveled are definitely affected by changes in initial velocity. It is important to note that any simulation that failed to maintain stable hover was not included in the average values, and the test conditions where all five of the repeated simulations were unsuccessful were left off of the plots altogether so that the results were not skewed by a failed test.

The most interesting part of Figure 6.10 is the nearly linear relationship between the reference model rise time and the maximum downrange distance traveled by the aircraft. Just like in Figure 6.2, there are small increases in distance traveled as approach speed is increased for a given rise time, but the additional linear increases in distance as rise time increases has the potential to make a reference model based transition very useful. As seen in Figure 6.6, the aircraft usually reaches the maximum downrange distance at the same time as it reaches 90 degrees pitch angle, where the transition is completed and stable hovering begins. The fact that the distance shares a linear relationship with both the approach speed and reference model rise time means

that the UAV operator could potentially use Figure 6.10 to determine how far away a transition needs to be initiated for things like perching or precision surveillance. Both of these applications would likely require the addition of actual waypoint control when in hover orientation to be practical, but the fact that the aircraft's distance traveled has the potential to be this predictable, even when the aircraft's attitude and altitude are the only things being controlled, is a very interesting and useful result.

The inconsistency of the altitude change for the longer duration transitions is directly related to the decrease is probability of success because of the extended periods of time where the aircraft is at very high pitch angles (between 20 and 85 degrees) and very low speeds, without fully being in hover orientation. The aircraft's attitude is largely at the mercy of the wind when in these high pitch conditions, so the divergence throttle needs to be used early and often to help the simulated aircraft track the reference model. The long periods of excess thrust mean that the aircraft keeps climbing as it tries to correct the yaw and pitch errors. The altitude change results are fairly linear for rise times between 1 and 5 seconds, however, so the same concept of being able to predict the final position of the aircraft could be applied to the altitude just like it was to the distance. These results are just based on simulations though so the possibility of actually using data like this to perform pinpoint transition maneuvers needs should be based on flight test results which are presented in Chapter 7.

Overall, the addition of a reference model input to the attitude PID controllers proved to be successful in simulations. Very short rise time reference models allowed for similar results to a step response but helped reduce the stress on the aircraft by smoothing and slightly slowing the initial pitch up at the beginning of the transition. The small amount of overshoot that was built into the reference models also helped the aircraft stop its forward momentum once stable hover was achieved and resulted in a more stable starting point for the hover that helped reduce the need for the

hover divergence throttle control. The HOVER_PID_REF simulations also showed the potential for precision aircraft placement at the end of the transition maneuvers even if the aircraft's position is not controlled directly during the transition, but flight test results had to be examined to see if this was actually feasible in a real aircraft.

6.3 Model Reference Adaptive Control

Before initial simulation testing, the HOVER_ADAPTIVE flight mode was designed to use Model Reference Adaptive Control during both the transitional and stable hover portions of the simulations. As discussed in Chapter 4, it was quickly discovered that MRAC was not well suited for maintaining hover because of the way that the adaptive gains continue to grow in magnitude any time the output error is non-zero. Since the airplane is constantly fighting wind disturbances and gravity while hovering, the system is rarely tracking the reference model perfectly which means that the adaptive gains continue to grow throughout the flight. This would eventually cause the gains to saturate to the point where even very small Euler errors would cause the control surfaces to be deflected to their maximum values. This is demonstrated in Figure 6.11 which shows an example of the time response of the system when using MRAC throughout the simulation, and in Figure 6.12 which shows he accompanying control surface deflections.

While the simulation shown in Figure 6.11 technically would have been considered successful because it didn't diverge until after the 15 second mark, the plots show that the airplane starts to undergo rapid oscillations in both the pitch and yaw axes between the 15 and 20 second marks. These oscillations, and the eventual yaw divergence they cause, are due to the growing magnitude of the control surface deflections, shown in Figure 6.12. Essentially, the adaptive gains grow to the point where the adaptive controller starts to work like a simple proportional controller in which the

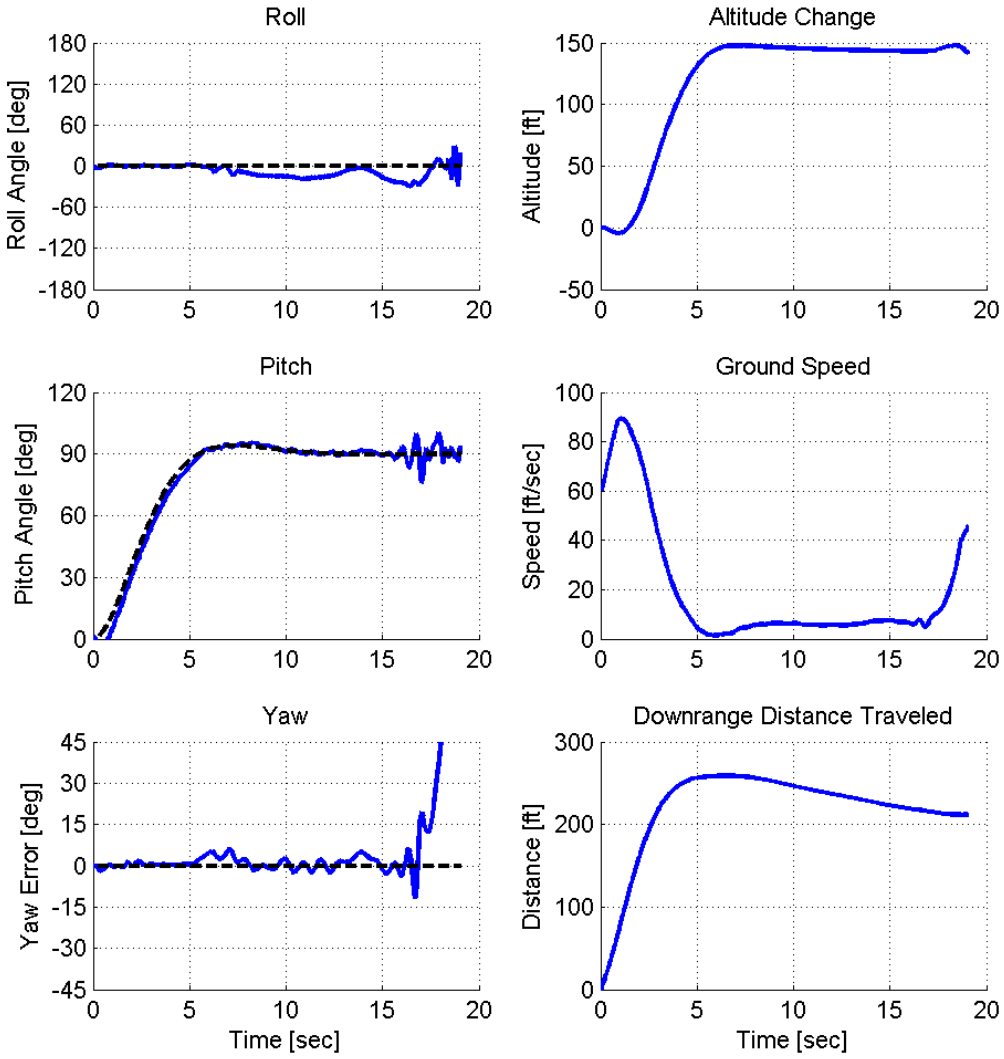


Figure 6.11: MRAC Problems In Hover: HOVER_ADAPTIVE simulation results (rise time = 3 sec, approach speed = 60 ft/sec, wind speed = 10 ft/sec) where MRAC is used throughout the simulation instead of just during the transition portion.

proportional gain is too high for the system and the gain margin is exhausted. As the adaptive gains and the control surface deflections start to grow due to the wind disturbances, the attitude response of the aircraft starts to become out of phase with the control surface deflections. The out of phase controllers then command even greater

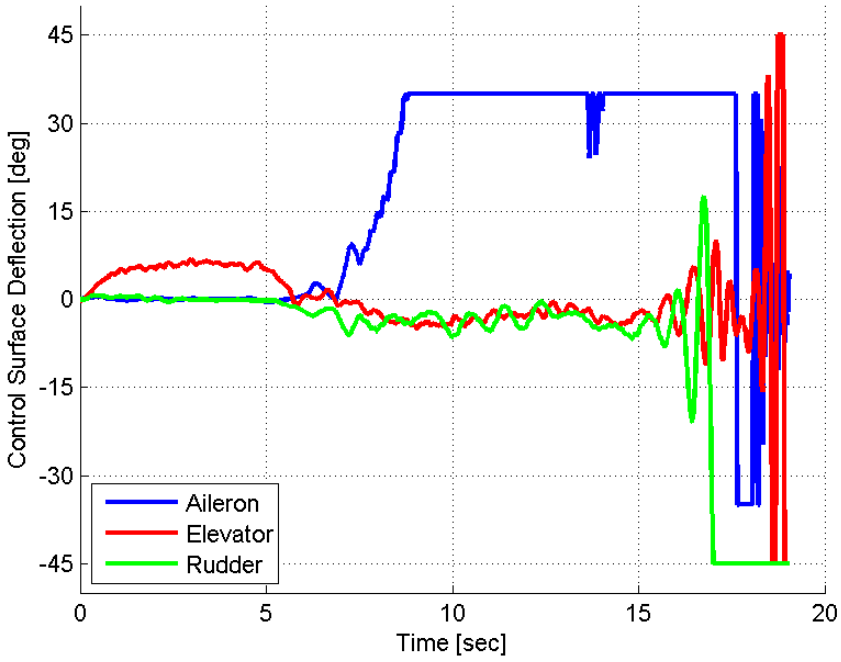


Figure 6.12: MRAC Control Surface Deflections In Hover

deflections to try and correct the Euler errors which only compounds the problem until eventually the oscillations and Euler errors get so large that there is no way for the system to recover and stable hover is lost. To ensure that this problem was not just a characteristic of the simulation, initial flight tests, where MRAC was used throughout the flight, were also conducted but also showed the same results. The flight test aircraft was able to complete the transition and even hold a very precise hover for a few seconds, but just like in the simulations, the aircraft quickly lost control when they adaptive gains grew too large.

This gain growth problem is why, as discussed in Chapter 4, MRAC is only used for the transitional portion of the flight (when pitch is less than 85 degrees) and then the attitude PID controllers with the hover flight gains are used to maintain the hover orientation. While MRAC was very effective during the transitional portion of the flight, which will be discussed shortly, it simply was not a good fit for maintaining

hover because of the way the algorithm works. A different formulation of adaptive control, perhaps one where the adaptive gains are limited to maximum values or have some way of decreasing in magnitude when the system is tracking the reference model properly, may be a better solution but examining these other types of controllers was out of the scope of this thesis.

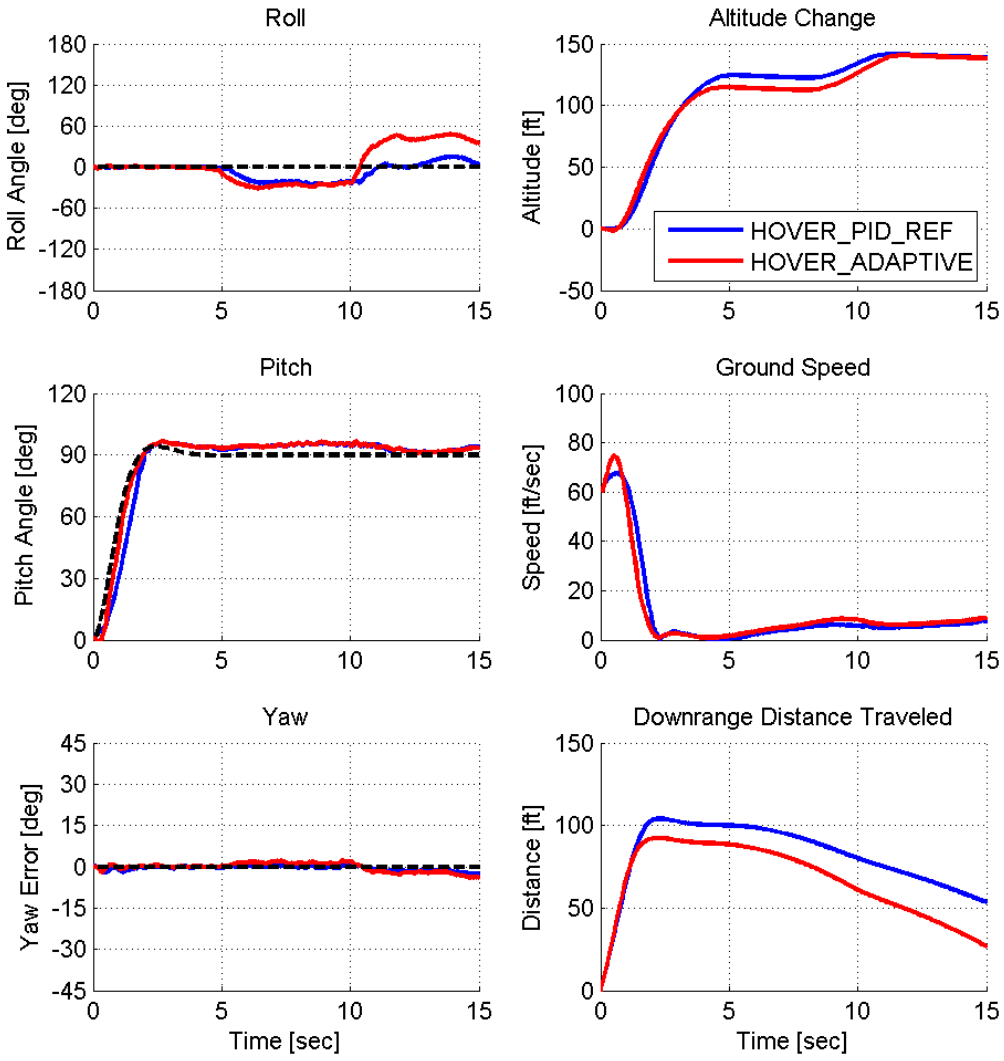


Figure 6.13: MRAC Simulation Example: HOVER_ADAPTIVE simulation results (rise time = 1 sec, approach speed = 60 ft/sec, wind speed = 10 ft/sec) compared to the analogous HOVER_PID_REF simulation.

Figure 6.13 shows an example of the HOVER_ADAPTIVE simulation results compared to the analogous HOVER_PID_REF simulation where the reference model rise time = 1 sec, approach speed = 60 ft/sec, and wind speed = 10 ft/sec (the same conditions where the HOVER_PID_REF results were compared to the step response in Figure 6.4). These plots serve as an initial proof of concept for the use of Model Reference Adaptive Control for fixed-wing transition-to-hover since the MRAC was able to successfully control the aircraft during the transition portion of the simulation and then hand off control to the attitude PID controllers once the pitch angle exceeded 85 degrees.

During the transition, however, the MRAC actually does a better job of tracking the pitch angle reference model than the attitude PID controllers do. As a result, the divergence throttle does not need to be used during the transitional portion of the simulation which means that the aircraft propeller produces a lower average thrust over that period of time. The result is that both the initial altitude gain and the maximum distance traveled are slightly lower for the HOVER_ADAPTIVE flight mode than they were for the HOVER_PID_REF mode since less kinetic energy was added to the system during the transition. This means that using MRAC to control the transition-to-hover has the potential to decrease the amount of space needed to complete the maneuver which would be an advantage for applications like flying in crowded urban environments.

A disadvantage of the HOVER_ADAPTIVE flight mode is that it suffers from the same initial speed increase caused by the throttle PID controller trying to maintain altitude at low pitch angles as the HOVER_PID_REF mode did. The speed increase in Figure 6.13 is even greater for the HOVER_ADAPTIVE results because, while the integrator term of the elevator PID controller carries over its trimmed value needed to maintain level flight when the transition is initialized, the adaptive gains are reset back to the their user defined initial values every time a new transition is started.

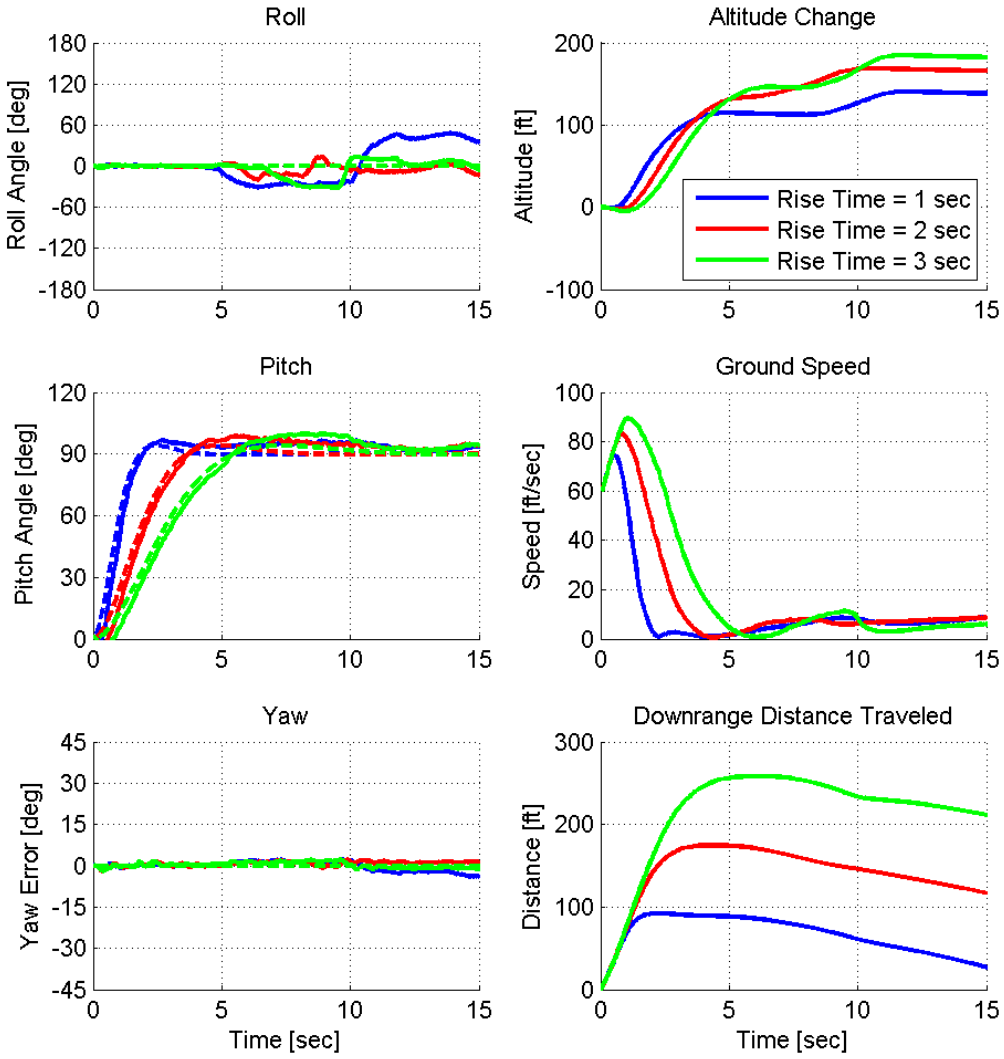


Figure 6.14: MRAC Simulation Initial Acceleration Problem

The period of time it takes for the MRAC to first measure that the aircraft's nose has started to drop and then apply the appropriate amount of elevator input to start the pitch up portion of the transition is actually enough for the airplane to loose a little bit of altitude and cause a large (almost 20 ft/sec) increase in velocity. This problem is magnified when the reference model rise time is increased, as can be seen in Figure 6.14. Unlike the HOVER_PID_REF flight mode, the initial acceleration problem for

the HOVER_ADAPTIVE mode was not solved by simply increasing the approach speed, as seen in Figure 6.15. Again, this is because the adaptive gains are reset to their user defined initial conditions at the beginning of every transition instead of carrying over the trimmed elevator deflection like the HOVER_PID_REF mode does.

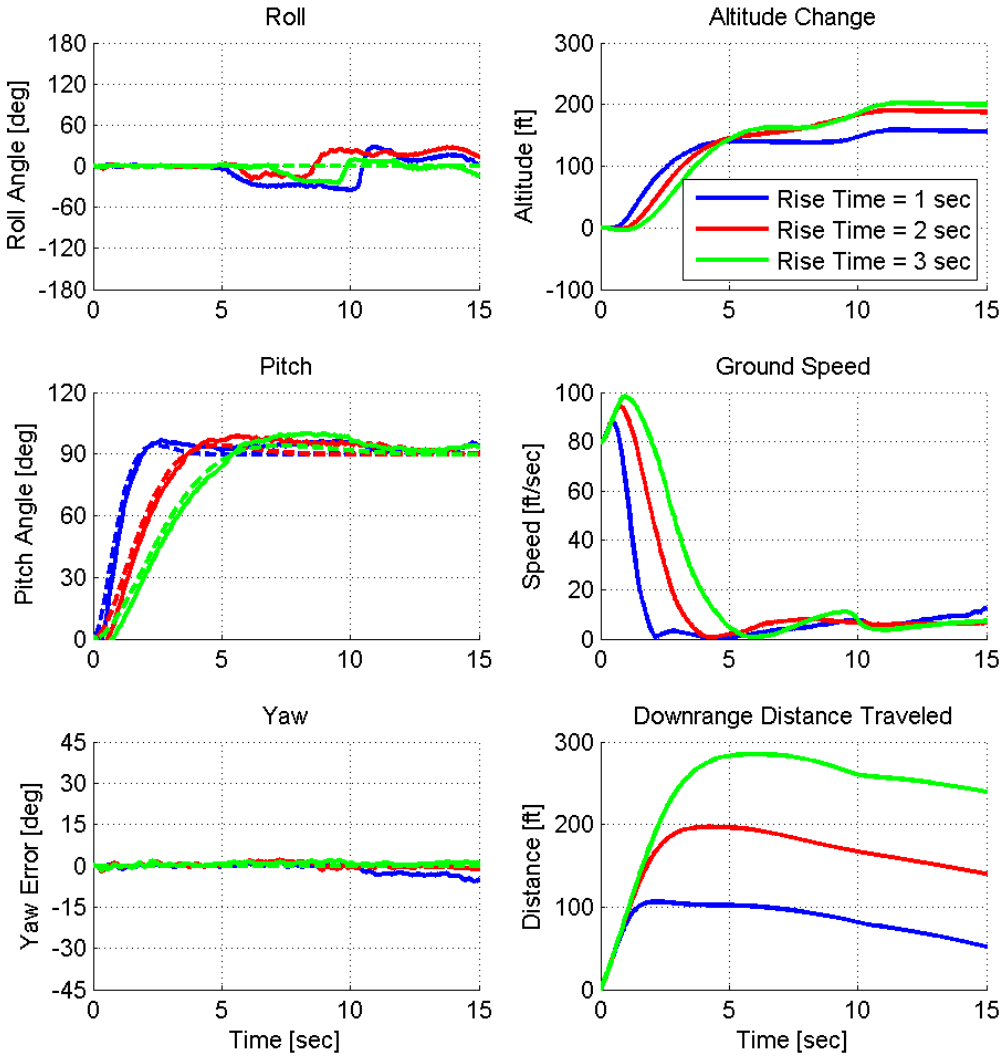


Figure 6.15: MRAC Simulation Acceleration Problem Unaffected By Approach Speed

The HOVER_ADAPTIVE simulation results showed the same trends as the other two hover flight modes in that the probability of success was not affected by changes in

MRAC Simulations: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
5	80%	60%	80%	100%	80%
7	40%	20%	40%	0%	25%
10	40%	20%	40%	20%	30%
15	20%	0%	0%	0%	5%
20	0%	20%	0%	0%	5%
Cumulative	60%	53%	58%	53%	56%

Table 6.2: MRAC Simulations: Probability of Success

the approach speed, as shown in Table 6.2. The large increases in initial speed should not cause any changes in the effectiveness of the MRAC, but may cause excess altitude and distance increases which would be undesirable in crowded airspace. Potentially, the ArduPlane firmware could be changed so that the level flight trimmed position of the elevator was carried over to the MRAC in the same way that it carries over to the attitude PID controllers. That way, the aircraft would not start to nose over and lose altitude momentarily when the HOVER_ADAPTIVE flight mode is engaged. This would require heavy modification to the firmware however, and was out of the scope of this thesis.

Table 6.2 shows that the HOVER_ADAPTIVE simulations performed very similarly to the HOVER_PID_REF simulations in terms of probability of success. Both flight modes successfully completed all tests with a reference model rise time of three seconds or less, and then started having problems with transitions that were longer than that. Interestingly, most of the HOVER_ADAPTIVE failures that occurred during longer duration transitions happened after the MRAC had handed off control of the aircraft to the attitude PID controllers, and then the PID controllers experienced

the same yaw divergence problem that was discussed in Section 6.2. An example of this is shown in Figure 6.16 where all four of the simulations shown were run with approach speeds of 60 ft/sec and wind speeds of 10 ft/sec, but the three faster transitions all successfully make it past 85 degrees pitch angle (where the attitude PID controllers take over), and it is only the longest duration transition that fails while the MRAC is still in control of the aircraft. The simulation that did fail while the MRAC was still in control shows the same rapid oscillations in the yaw and pitch axes that were seen in Figure 6.11, right before that simulation failed, which shows that the same saturation of the adaptive control gains occurred.

Only 20 of the 71 unsuccessful HOVER_ADAPTIVE simulations diverged, or at least started to diverge, while the MRAC was still in control of the aircraft (some crossed the 85 degree pitch angle threshold while in the late stages of the MRAC induced oscillations at which point it was too late for the PID controllers to recover when they were engaged). Even more interesting is the fact that 16 of those 20 MRAC caused failures occurred during simulations where the reference model rise time was 20 seconds. That means that only 4 of the remaining 55 HOVER_ADAPTIVE simulations failures (7%) were caused during the MRAC controlled transition-to-hover instead of happening later on while the PID controllers tried to maintain stable hover. This is further proof that Model Reference Adaptive Control is a potentially valid solution to controlling the nonlinear transition-to-hover maneuver for a fixed-wing aircraft.

Figure 6.17 shows the altitude change and distance traveled trends for the HOVER_ADAPTIVE simulations as a function of the reference model rise time. The HOVER_ADAPTIVE simulations showed the same linear relationship between the reference model rise time and the average maximum distance the aircraft traveled downrange as the HOVER_PID_REF simulations did. The HOVER_ADAPTIVE flight mode did seem to be much more effective at limiting the altitude gain during the longer

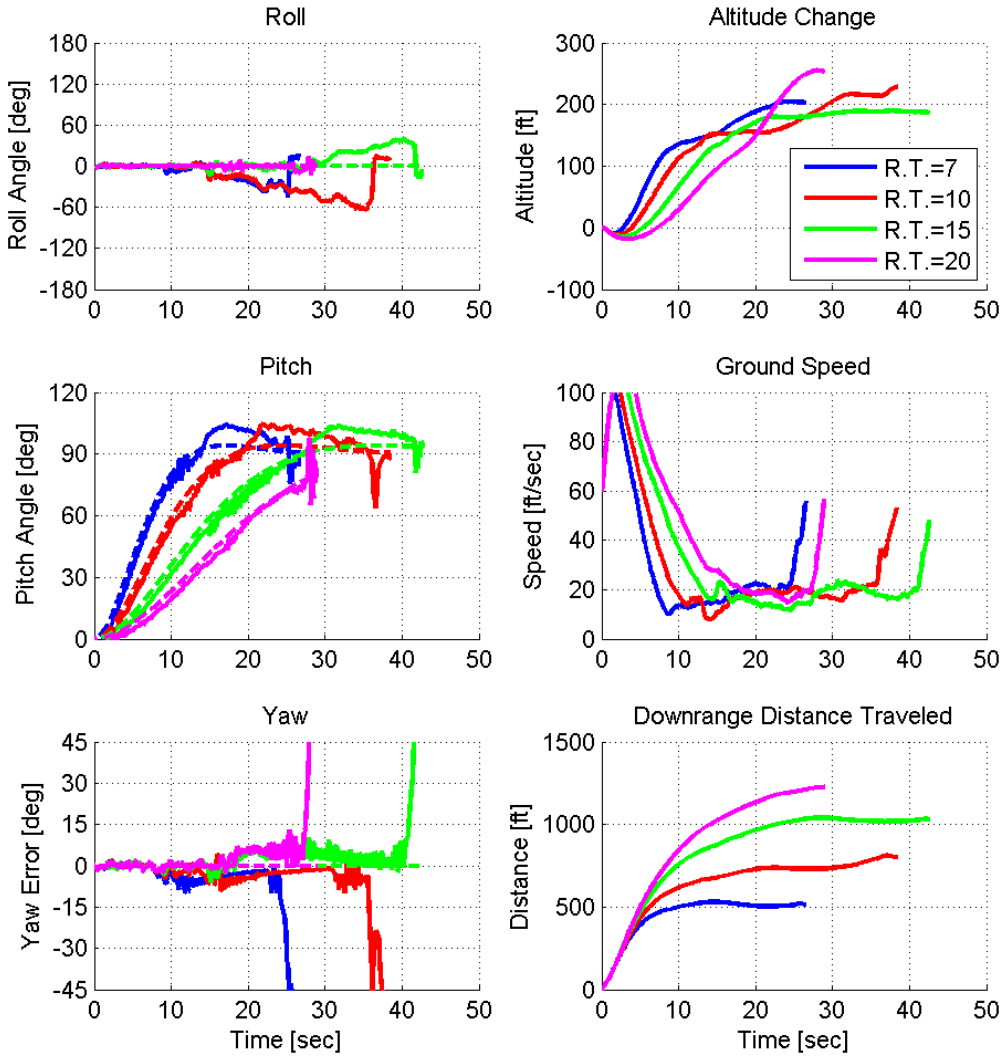


Figure 6.16: MRAC Simulation Failures

transitions however as almost all of the simulated tests showed total altitude changes near or less than 200 feet. The increased effectiveness of the MRAC transitions becomes even more apparent when the altitude and distance trends for both the HOVER_ADAPTIVE and HOVER_PID_REF flight modes are plotted on top of each other like they are in Figure 6.18. The altitude change trends are nearly identical for the two flight modes when the reference model rise time is below 5 seconds, but

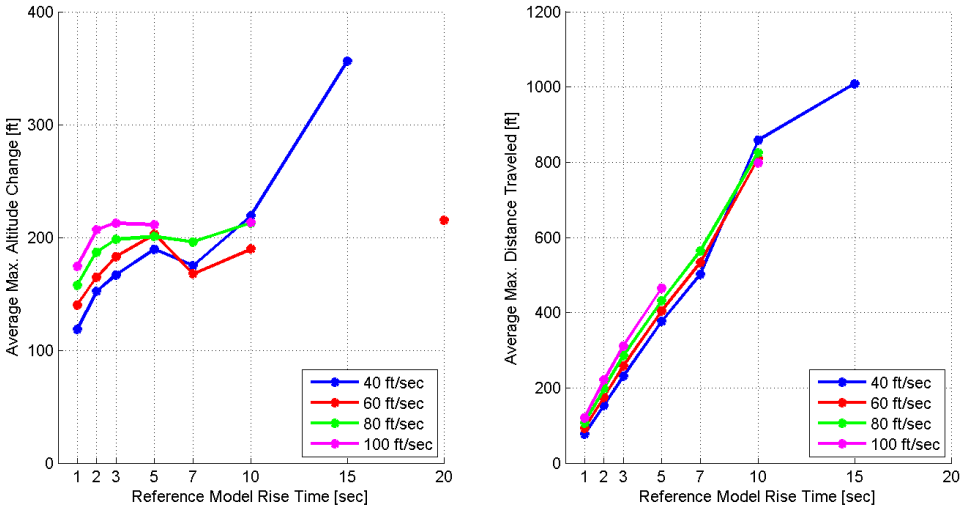


Figure 6.17: MRAC Simulation Results: HOVER_ADAPTIVE simulations showed the same linear relationship between reference model rise time and the distance the aircraft travels as compared to the HOVER_PID_REF simulations, but the altitude change was fairly consistent even as the transitions were lengthened.

the intermediate length MRAC simulations experience much less altitude gain than their PID controlled counterparts. This is likely due to the MRAC's increases ability to closely track the input reference model during the highly nonlinear portions of the transition when the aircraft is at very low speeds and very high pitch angles. Tracking the reference model very closely based on control surface deflections alone means that the hover divergence throttle logic is not needed as often when in the HOVER_ADAPTIVE mode as it is in the HOVER_PID_REF mode during long transitions. This means less kinetic energy is added to the system during the transition and as a result, less altitude is gained over the entire duration of the maneuver.

Overall, the HOVER_ADAPTIVE simulations proved that Model Reference Adaptive Control is a feasible solution for controlling fixed-wing transition-to-hover maneuvers. It produced very similar results to the PID controlled maneuvers with respect to both the probability of success and the distance the aircraft travels before reaching

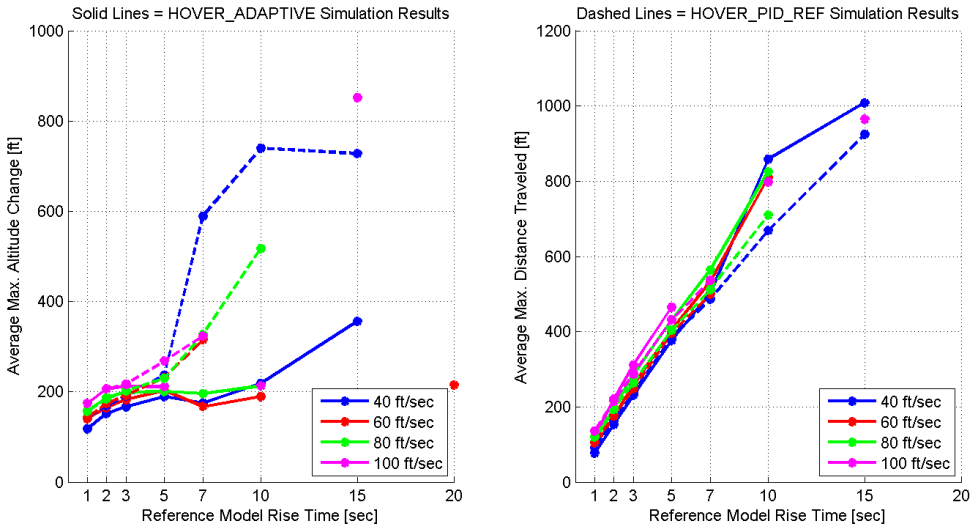


Figure 6.18: MRAC Simulation Results Compared to PID Control Reference Model Response

the hover orientation. The adaptive control gains in the MRAC algorithm did seem to maintain precise control of the aircraft's attitude better during the highly non-linear portions of the transition when compared to the simple PID controllers. This resulted in the aircraft generally climbing less during the transitions since the autopilot did not need to add as much throttle input to keep the Euler errors within reasonable limits. Since almost all of the failures seen in the simulations occurred after the aircraft had already been hovering for a period of time, the MRAC algorithm was usually not to blame for the lack of success. Minimizing the altitude gained during transitions could be very advantageous for applications like vertical landings, though, where minimizing the time spend in hover, and thus the risk of losing control, would be highly desired.

6.4 Hover-to-Level Transition

Simulations of the hover-to-level flight transitions were conducted by initializing the simulation with the aircraft at roll, pitch, and yaw angles of 0, 90 and 0 degrees

respectively. The aircraft was allowed to hold stable hover for a few seconds until a step response was used to change the desired pitch angle from 90 degrees to 0 degrees, which is the same thing that happens when the autopilot is switched from one of the three hover flight modes back into STABILIZE mode. Roll angle was attempted to be held at zero throughout the transition back to level flight, but the rudder is unused when in STABILIZE mode so any changes in the yaw error were due to a non-zero roll angle causing the aircraft to turn slightly.

As mentioned in Chapter 4 the throttle is controlled manually by the pilot when the autopilot is in STABILIZE mode so it was set to 75% throughout the transition for consistency. Since all of the hover-to-level simulations have the same stationary hover starting conditions, each is tracking a step input, and the throttle setting is always the same, the simulations were only repeated five times in order to monitor the maneuver in various wind conditions. Just like the transition-to-hover simulations, the hover-to-level simulations were tested at 0, 5, 10, 15 and 20 ft/sec wind speeds. The results from the 10 ft/sec wind speed simulation are shown in Figure 6.19. The initial portion of the simulation where the aircraft was only holding stable hover was not included in the results plots.

All five of the attempted simulations were successful in returning the aircraft back to steady level flight. Figure 6.19 shows that the elevator is able to quickly pitch the nose of the plane down and actually overshoots the desired 0 degree pitch angle by about 15 degrees. This overshoot is due to the fact that the aircraft still has very little airspeed at that point in the simulation, and as a result is unable to stop the rotation of the aircraft in time to prevent the overshoot. Because of this overshoot, the aircraft quickly starts to pick up speed due to gravity and the thrust of the propeller acting together. After losing about 40 feet of altitude, the aircraft reaches its final steady state velocity at which point it has enough airspeed to maintain altitude. The aircraft then starts to climb again for the final 5 seconds of the simulation because of

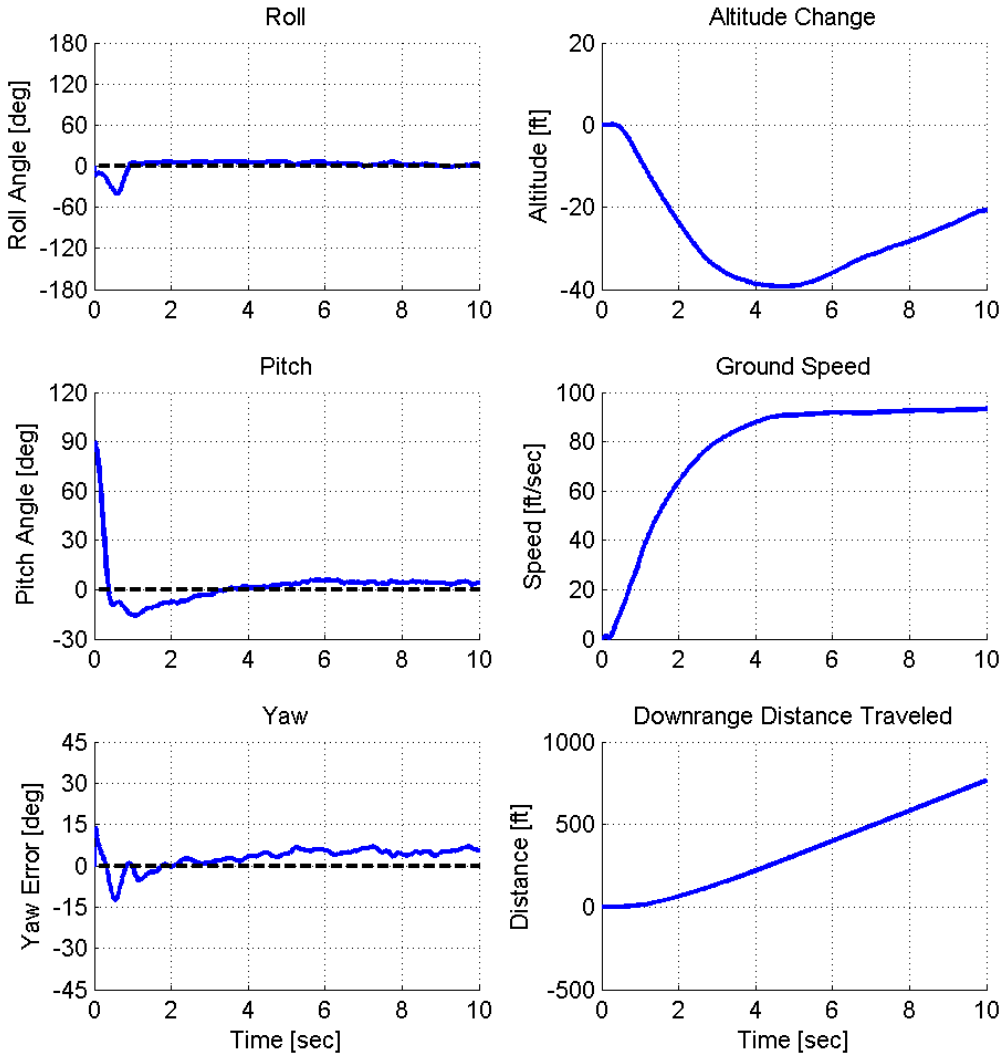


Figure 6.19: Hover-to-Level Simulation Example

the integrator windup that occurred during the initial portion of the simulation. The altitude increase would dissipate after a few seconds, though, once the steady state pitch error was eliminated.

All five of the hover-to-level simulations showed very similar results, even with the varying simulated wind conditions. The average altitude loss at the beginning of the maneuver was 39.8 feet and the standard deviation was only 0.9 feet. Each simulation

showed the same initial overshoot in the pitch response of the aircraft because of the lack of airspeed over the elevator when the aircraft first reaches level orientation, but all of the simulations also showed the same ability for the controllers to quickly bring the nose of the airplane back to level and to hold the roll constant throughout the simulations.

Most importantly, the simulations showed that the autopilot should be able to transition from hover back to level flight without any additional modifications to the firmware. By switching from one of the three hover flight modes directly into any of the standard ArduPlane level flight modes, the airplane should smoothly transition back to level flight and continue its mission without putting any extreme stress on the airframe or any payloads on board.

7 FLIGHT TEST RESULTS

Actual flight test results for the three different hover flight modes are presented here. The same 68 combinations of controller type, reference model rise times and maneuver approach speeds that were simulated and discussed in Chapter 6, were repeated as closely as possible in flight with the Carbon-Z Yak 54 airframe and ArduPilot-Mega 2.5 autopilot described in Chapter 2. Each unique flight test was then repeated 5 times, just like the simulations, for a total of 340 separate test. There were, however, a few minor differences between the flight tests and simulations due to the difficulties of flying a real aircraft.

First, no real attempt was made to ensure that each real transition-to-hover maneuver started at 100 feet off the ground like the simulations did. Doing so would have needlessly increased the pilot's workload since relative altitude change is all that really mattered for the test results, as opposed to the absolute altitude of the aircraft. Most transitions were started with the aircraft between 100 and 300 feet off the ground to ensure the aircraft was always at a safe altitude, but still close enough for he pilot to maintain good visual contact.

Secondly, no attempt was made to wait for wind conditions that matched either the wind direction or wind speeds used in the simulations. Waiting for days where the wind was blowing directly north to south and at exactly 0, 5, 10, 15, or 20 ft/sec would have meant that the 340 separate tests could potentially have taken years to complete. During each flight test, the wind speed was simply measured with a hand-held anemometer and the aircraft was flown directly into the wind as long as it was safe to do so. The autopilot firmware was set to try to hold the initial heading of the real aircraft during the transition, whatever it happened to be, rather than trying

to hold due north like the simulated controller did in the simulations. To ensure the safety of both the aircraft and the ground crew, no flight tests were attempted when the wind speed exceeded 20 ft/sec.

The last difference between the flight test and simulation conditions is that the approach speed for the flight tests was sometimes not exactly equal to the desired value. The way that each flight test was run involved an assistant reading off the aircraft's ground speed from the Mission Planner telemetry link while the pilot tried to make small changes to the throttle in order to achieve the desired speed. Sometimes, during the short period of time it took the pilot to change flight mode, wind gusts or continued acceleration/deceleration of the aircraft would cause the speed to change slightly before the transition could be initiated. The wind played an especially large roll in the 100 ft/sec approach speed flight tests since the aircraft required the throttle be set to 100% even in calm winds to reach the desired speed. Since the Mission Planner software only reports the aircraft's GPS based ground speed, a strong head wind sometimes meant it was impossible for the aircraft to ever reach the desired 100 ft/sec ground speed even at full throttle. The results in this chapter are presented as $\sim$100 ft/sec for the highest velocity tests since the aircraft often did not fully reach the desired approach speed and the transition was started once the aircraft had reached its maximum steady state ground speed at full throttle.

As a reminder, the GPS ground speed is calculated as the magnitude of the aircraft's speed, regardless of direction, so the the ground speed will never drop below 0 ft/sec in any of the plots presented in this chapter. Also, the yaw error, altitude change, and distance traveled plots are all calculated relative to the initial heading and position of the aircraft so that the flight test results are easily comparable to one another and to the simulation results in Chapter 6. Throttle controller input was also set to 0 ft/sec descent/climb rate for the entire duration of the flight tests the same way it was for the simulations.

The success or failure of each flight test was also dependent on whether the aircraft was able to transition to and then maintain stable hover for at least 15 seconds (measured from the time the transition is initiated), or a time equal to the reference model rise time multiplied by five, the same as it was for the simulations. Determining whether the aircraft had diverged from the desired attitude was done by the pilot, however, instead of using the 45 degree Euler error rule. If at any time during the test the pilot felt that the autopilot had lost control of the aircraft and was not going to be able to recover, the pilot would switch the aircraft back to MANUAL flight mode and return the aircraft to a safe flying condition immediately.

Finally, a complete record of the data collected from each of the 340 flight tests is available in Appendix B.

7.1 PID Control Step Response

Flight tests of the HOVER_PID_STEP fight mode showed that the aircraft was able to transition to and maintain stable hover for 15 seconds for each of the 20 different tests. This was the same level of reliability that the HOVER_PID_STEP simulations showed. An example of the plotted results from one of the 60 ft/sec approach speed tests is shown in Figure 7.1 and is plotted together with the matching simulation from Figure 6.1 for comparison.

It is important to remember that the simulator developed for this thesis was never intended to accurately represent the nonlinear aerodynamics of a fixed-wing airplane transitioning from level to hover flight. It was simply developed as a design tool to help develop the control algorithms that were used in the actual autopilot, so the goal of the simulator was only to show similar trends as the flight test results, but was never expected to match the flight tests exactly. Figure 7.1 provides a good example for showing that the simulator was successful in doing this and also highlights a few

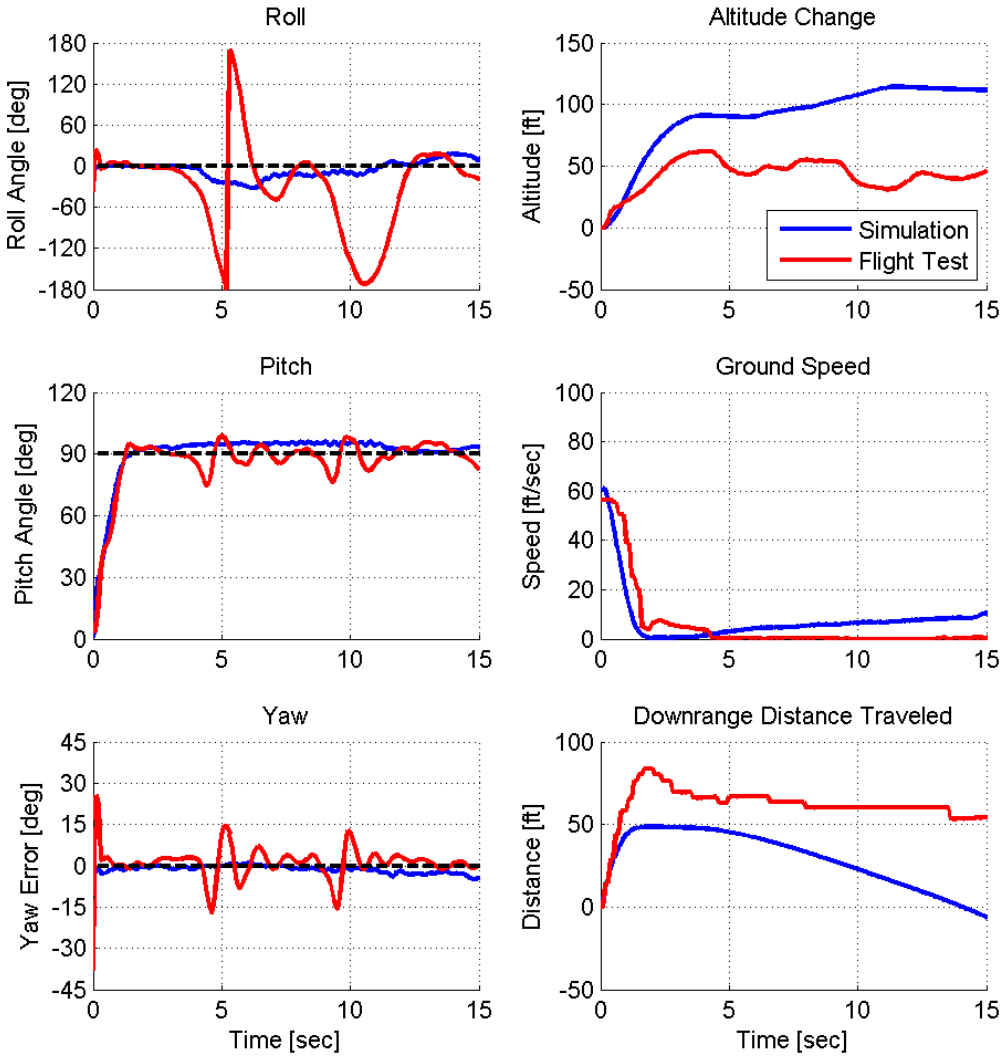


Figure 7.1: PID Control Step Response Flight Test Example: HOVER_PID_STEP flight results with approach peed = 60 ft/sec and wind speed = 4.3 ft/sec, compared to simulation results with matching conditions.

major differences between the flight tests and the simulations.

First off, the roll, pitch, and yaw responses of the simulated and actual aircraft match almost exactly during the first four seconds of the flight, which represent the transition-to-hover and the first few seconds of stable hover. This result was very

encouraging since it proved that the simulator was a good tool for developing the attitude control algorithms, at least for the shorter duration transitions. The one exception is the large variation in yaw error that occurs in the first second of the flight test results. This large variation in yaw angle at the very beginning of the transitions was seen throughout the flight test results but was never actually observed by the pilot or spectators in flight because it is caused by a sensor error and not the actual motion of the aircraft. DIY Drones refers to this phenomenon as the attitude sensors becoming "dizzy" when the aircraft is experiencing large angular velocities like the ones seen during the initial pitch up for the transition maneuver. This "dizziness" is caused by the way that the autopilot mixes the inertial navigation sensor readings and the magnetometer readings to correct any yaw drift errors, and while the inaccurate yaw angle does get reported to the PID controllers when the "dizziness" occurs, it happens for such a short period of time that the rudder response is very limited and was never observed to cause anything more that a very small disturbance in the actual yaw angle of the aircraft in flight.

After the aircraft had been hovering for a few seconds, the flight tests and simulations start to vary much more. The most glaring difference is the very large variations in roll angle that the actual aircraft experienced during flight tests. Just after the 5 second mark in Figure 7.1, the flight test aircraft actually makes a full 360 degree counter-clockwise rotation which is caused by the large motor torque created when the throttle is increased. This motor torque was difficult to capture and often underestimated in the simulator since none of the simulations ever showed roll angle variations greater than 60 degrees. Large motor torque induced roll errors, and even the full rotations, were the norm rather than the exceptions for flight tests however. While the excess rolling typically did not affect the stability of the hover, since the rudder and elevator usually did a good job of keeping the nose pointed straight up even as the aircraft was doing pirouettes in the sky, the inability to keep the aircraft's belly

pointed in one direction while hovering dramatically decreases the aircraft effectiveness for stationary surveillance applications. The motor torque problem is the main reason many of the VTOL aircraft mentioned in Chapter 1 either have two motors that spin in opposite directions, or one motor with contra-rotating propellers. Since this thesis was focused on the feasibility of developing a transition-to-hover controller for a standard style fixed-wing aircraft however, the roll problem simply had to be accepted as a limitation of the airframe.

Another less drastic, but still important, difference between the flight tests and simulations while the aircraft is hovering is that the yaw and pitch variation tended to be greater during actual flights than it did during simulations. Figure 7.1 shows that the pitch and yaw errors in hover typically vary between ± 15 decrees for the flight test, while the simulation tends to stay between ± 7 degrees. Whether the flight test aircraft was actually varying its pitch and yaw angles this much or whether the errors were exaggerated because of the "dizziness" problem is impossible to tell, but since there were no other ways to measure the aircraft's attitude externally (like a motion capture system), the autopilot measurements had to be accepted as true.

Interestingly, the flight test showed that the aircraft actually did a better job of limiting altitude gain in real life than it did in the simulator. This is probably due to the fact that the parasite drag estimate used in the simulator was likely much too low, so the actual airframe bleeds off kinetic energy through drag much quicker than the simulated aircraft does. The actual aircraft was able to maintain constant altitude fairly well throughout the hover though. The bobbing up and down that did occur is probably due to the fact that the autopilot uses a combination of the barometer and GPS based altitude readings to determine the aircraft's altitude. This mixing can sometimes create a small amount of lag in the system which would account for the more violent jumps in the flight test altitude measurement when compared to the smooth changes in the simulation altitude. The same reasoning also applies

to the rather discrete looking distance traveled plot for the flight test, since that measurement is based on the 10 Hz GPS readings that have the tendency to miss updates occasionally.

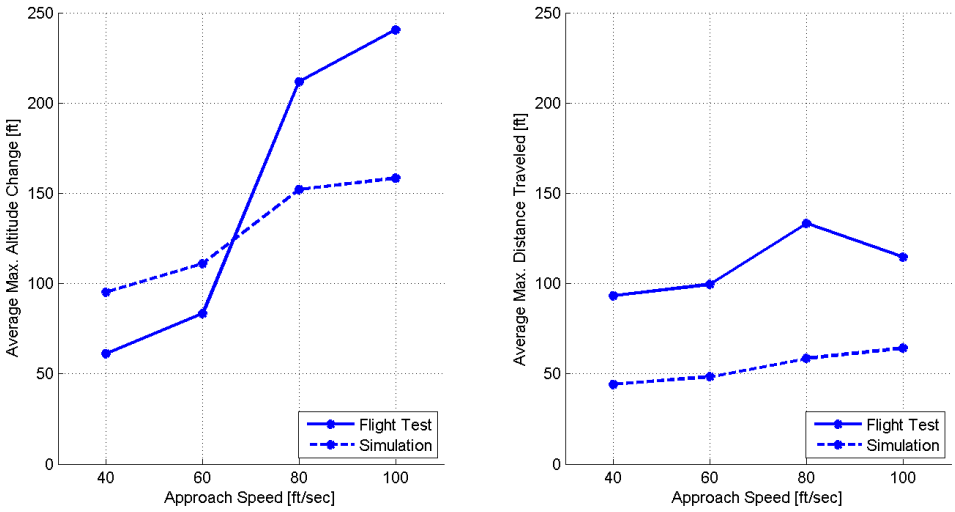


Figure 7.2: PID Control Step Response Flight Test Results: HOVER_PID_STEP average altitude change and average distance traveled both increase as a function of the aircraft's initial speed, just like the simulated aircraft, but the flight test values were different.

Figure 7.2 shows the average altitude change and average distance traveled by the aircraft as a function of the approach speed for the HOVER_PID_STEP flight mode, and compares the flight test results to those of the simulations. While the actual values did not match between the simulations and the flight tests, the trends did in that the altitude change shares a quadratic relationship with the approach speed and the distance traveled is more linear. The fact that the flight test distance trend is almost identical to the simulation trend but just at higher values likely just means that the minimum turning radius of the simulated aircraft was smaller than that of the real aircraft, but that the two have a similar ability to control the attitude of the aircraft during the transition, as was shown in Figure 7.1. As mentioned previously, the differences between the flight test and simulation altitude change is likely due to

improper modeling of the aircraft's drag in the simulations.

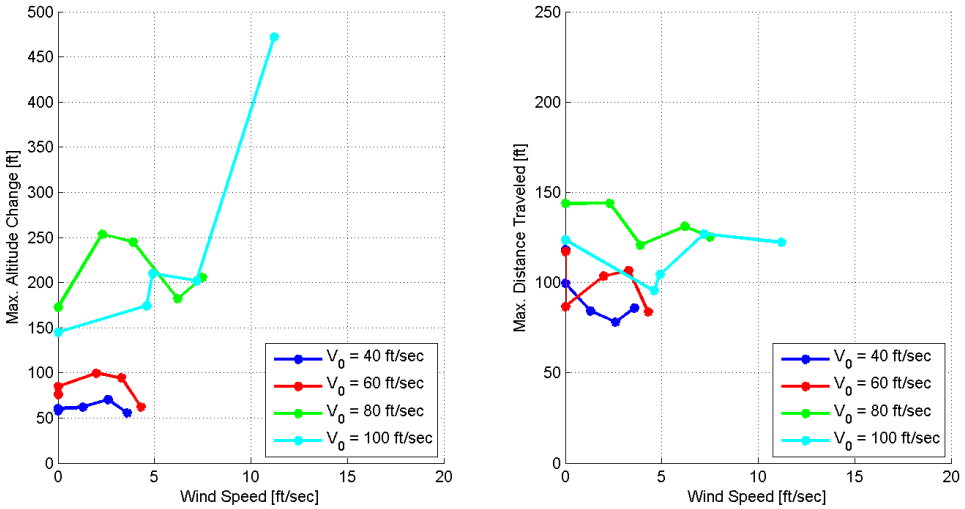


Figure 7.3: PID Control Step Response Flight Test Wind Sensitivity

Figure 7.3 shows how the altitude and distance traveled changed with the wind speed for the HOVER_PID_STEP flight tests. These results were similar to the ones shown for the simulations in Figure 6.3 in that there was no consistent and predictable relationship between the wind speed and either the altitude change or the distance traveled by the aircraft. Figure 7.3 does show that the the flight test results were much less consistent than the simulation results shown in Figure 6.3, however. This was not a surprise though, as flight test results are almost always less consistent than simulations because of factors like wind variability, real time hardware problems, and accidental pilot influence.

The HOVER_PID_STEP flight tests, first and foremost, served as a proof of concept for the attitude and throttle controllers that were designed using the simulator. All 20 of the attempted step response flight tests were successful which shows that the quaternion based Euler angle error discussed in Chapter 3 and the discrete attitude PID controllers described in Chapter 4 are capable of working together to control the transition-to-hover of a real fixed-wing aircraft with a relatively simple and cheap

commercially available RC autopilot.

One drawback of commanding the transition through a step input, that was not as obvious during the simulations but became very apparent during the flights, was that the step response of the real aircraft was very violent and put a lot of stress on the airframe. When the transition was initiated, the initial pitch up of the aircraft was very fast and put a large amount of strain on the wings of the aircraft. Luckily the Yak 54 model used in the flight tests is designed to be able to handle such high-G maneuvers but other airframes, and possibly sensitive payloads, might be damaged by such a maneuver. Luckily, the HOVER_PID_REF and HOVER_ADAPTIVE flight tests showed that using a reference model input, instead of step function, drastically reduced the strain on the airframe.

7.2 PID Control Reference Model Response

The HOVER_PID_REF flight mode was designed to maintain the same level of reliability and ease of use that was seen with the HOVER_PID_STEP flight mode, but also give the UAV operator the option of changing the amount of time it takes the aircraft to transition from level to hover flight. Simulations showed that the HOVER_PID_REF mode was only expected to successfully complete 53% of the attempted transition tests, but further examination revealed that the relatively high failure rate may have been due to deficiencies in the physics of the simulations and not deficiencies in the controllers themselves. Simulations also showed however, that potentially the altitude change and downrange distance traveled by the aircraft during the transition maneuver could be indirectly controlled by modifying the input reference model rise time of the HOVER_PID_REF mode. At the very least, the HOVER_PID_REF is expected to make the transition-to-hover smoother than was seen during the HOVER_PID_STEP flight tests.

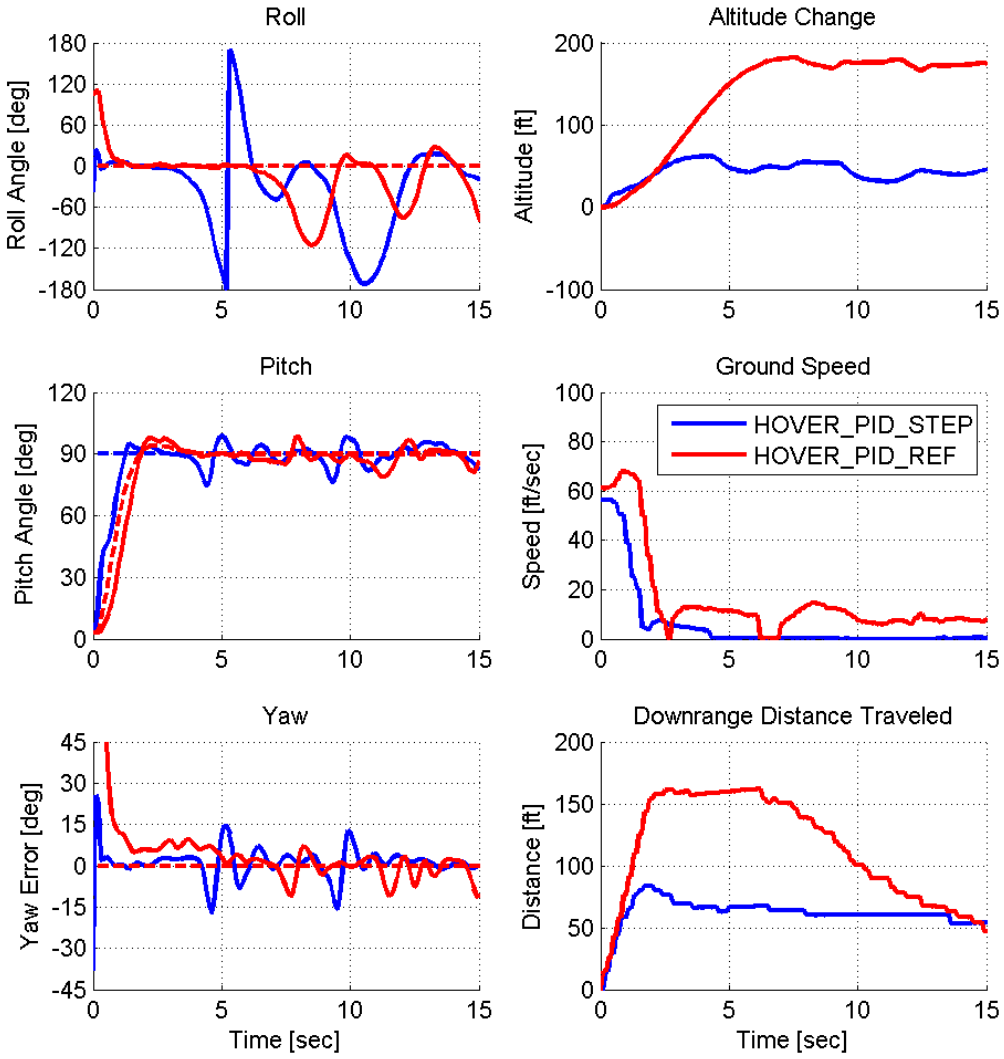


Figure 7.4: PID Control Reference Model Response Flight Test Example: HOVER_PID_REF flight results with approach speed = 60 ft/sec, wind speed = 4.6 ft/sec, and rise time = 1 sec, compared to analogous step response flight test (wind speed = 4.3 ft/sec).

Changing the PID controller input from a step function to a second order reference model yielded very similar results during flight tests as it did in simulations. The HOVER_PID_REF flight mode produced very similar results to the HOVER_PID_STEP mode for the shorter rise time reference models. An example

of this is shown in Figure 7.4 where the step response from Figure 7.1 is plotted with the results from one of the HOVER_PID_REF flight tests with the same approach speed and a reference model rise time of 1 second.

Just like in the simulations, the change to a reference model input significantly smoothed the initial pull up at the beginning of the transition compared to the HOVER_PID_STEP flight mode. Similarly, the HOVER_PID_REF flight test results show a small increase in ground speed at the beginning of the test due to the airplane trying to maintain altitude while at low pitch angles, which was also seen in simulations. Figure 7.4 does illustrate a major difference between the HOVER_PID_REF flight test and simulation results, however.

As mentioned in Chapter 6, the main cause of simulation failures was the aircraft's yaw angle diverging due to the lack of control effectiveness of the rudder when at low speeds and high pitch angles. Figure 7.4 shows that during the HOVER_PID_REF flight test the aircraft had a yaw error greater than 5 degrees for almost all of the first 5 seconds of the test. The cause of this initial yaw error is hard to tell. Potentially, the initial "dizziness" of the sensors lasted long enough and had enough influence on the PID controller inputs that a large rudder deflection was commanded during the initial pull up which caused the aircraft to be turned as the pitch angle increased. It is also just as likely, however, that the initial yaw error was caused by a wind gust and the sensors were merely measuring the error once the "dizziness" had dissipated. Without any external sensors to validate the measurements taken on board the autopilot, it is impossible to tell the true cause, but what is important is the fact that the flight test aircraft was able to recover from the large initial yaw error while the simulations so often were not.

Figure 7.4 shows that the throttle and rudder controllers reacted exactly as designed in order to stabilize the flight test aircraft. Because the yaw error was greater

than 5 degrees, the hover divergence throttle was triggered and the throttle was increased to 75%. This caused the aircraft to continue climbing and increasing airspeed over the control surfaces even after the aircraft had reached 90 degrees pitch angle. This can be seen in the altitude change plot of Figure 7.4, where the step response results shows the aircraft stopped climbing after about 50 feet because the yaw and pitch errors were small enough to not need the divergence throttle logic. The increase in propeller down-wash and overall airspeed helped the rudder stop the growth of the yaw error around 3 seconds into the test, and then the integrator term in the rudder PID controller slowly increased in magnitude until the remaining yaw error was eliminated about 6 seconds into the flight test. The divergence throttle logic was then deactivated and the throttle PID controller was able to maintain nearly constant altitude for the remainder of the flight.

Figure 7.5 illustrates the three different ways that the autopilot reacted to large yaw errors while the aircraft was hovering. All three of the flight test results in Figure 7.5 are meant to be repeats of the same test, where the approach speed is 40 ft/sec and the reference model rise time is 3 seconds. In fact, all three of the tests were conducted in succession so there were virtually no changes in wind conditions between the tests. The interesting thing about these three tests is that all of them experience a large yaw error just as the airplane reaches 90 degrees pitch angle but all three tests have very different outcomes. First, the blue results represent the test with the most desirable outcome where the attitude PID controllers quickly eliminate the yaw error and the divergence throttle logic was only needed for a very short period of time so minimal altitude change occurs. Next, the red results show a test where the rudder controller initially struggles to correct the yaw error and the divergence throttle is needed for a longer period of time, similar to the situation that was shown in Figure 7.4. The aircraft does eventually return to stable hover but it takes about 5 seconds to do so, which causes nearly 100 feet of excess altitude gain compared to the blue

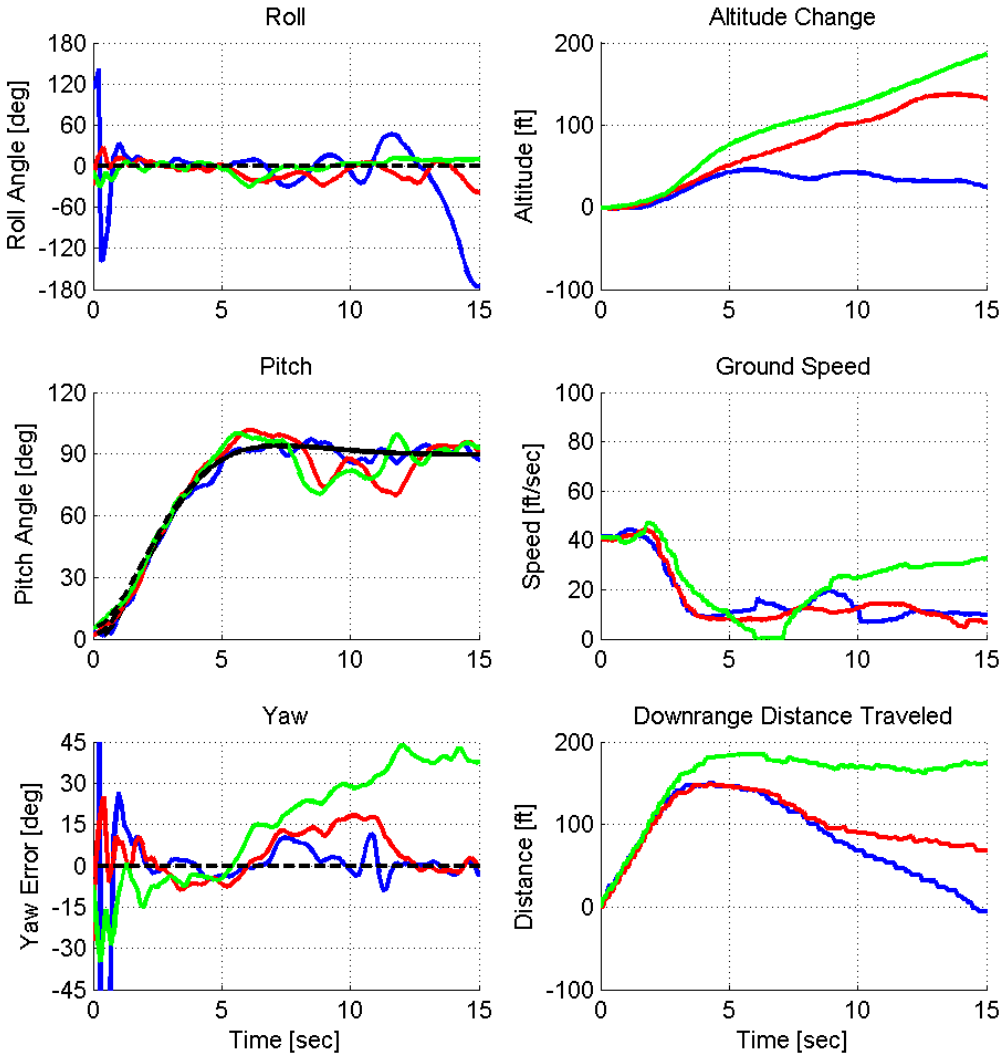


Figure 7.5: PID Control Reference Model Response Flight Test Variability

results. Finally, the green results show a situation where the rudder fails to ever correct the yaw error and the aircraft diverges in the same way that was seen so often in the simulation results.

One of the most interesting results of the HOVER_PID_REF flight tests was that, on a few occasions, the airplane actually stopped moving both forwards and upwards in the middle of the longer duration transition maneuvers and then proceeded to hold

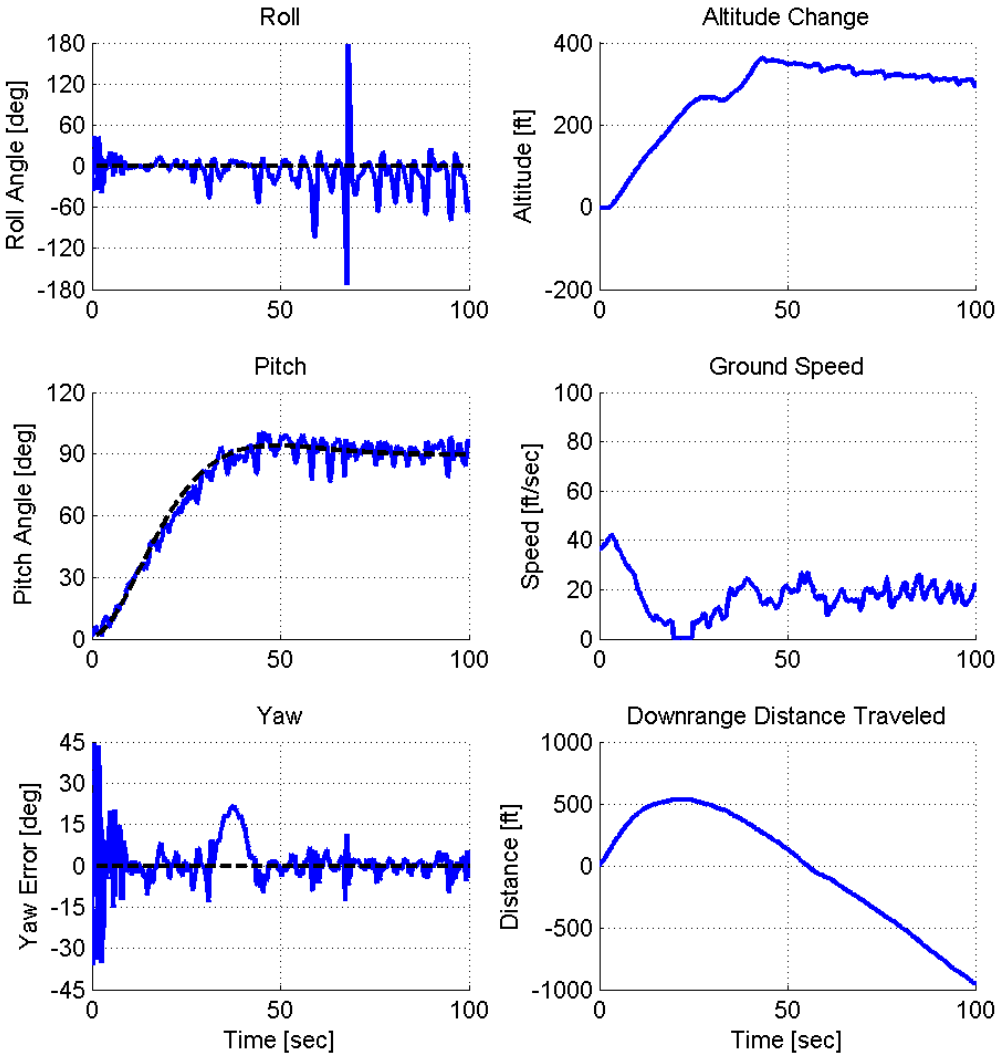


Figure 7.6: Long Duration PID Control Reference Model Response Flight Test

nearly constant position in the air while it slowly increased its pitch angle to the desired 90 degrees. This result looked very similar to the way that very talented RC pilots can transition an airplane from level flight to hover with hardly any altitude gain. An example of this is shown in Figure 7.6 where the airplane is holding nearly constant altitude and downrange distance between the 25 and 35 second marks even though the pitch angle undergoes a controlled increase of about 10 degrees during

that time.

Figure 7.6 also shows that the elevator controller and the throttle controller have to work much harder during the longer transitions than they do during the short rise time transitions. The pitch response shows that the aircraft undergoes small oscillations in the pitch angle throughout the transitional portion of the test that are caused by the aerodynamic pitching moment and the negative pitching moment from the forward C.G. of the aircraft constantly working against the elevator controller while it tries to slowly increase the pitch angle of the aircraft over a 40 second time period. These pitch oscillations were not seen during the shorter duration transitions because the aircraft had enough momentum to carry it through the entire short transition with enough airspeed so that the pitch angle increased smoothly instead of oscillating. Figure 7.6 proves that the closed loop PID controllers have the ability to control the aircraft through the non-linear aerodynamics that occur at low speeds and high angles of attack, instead of just relying on having enough kinetic energy to complete a simple pull up maneuver.

While Figure 7.5 is proof that there was much more variation in the results of the flight tests compared to the simulations, the hover controllers were actually much more successful in transitioning to and maintaining stable hover during actual flight tests than they were in simulations. Table 7.1 shows that 92.5% of the HOVER_PID_REF flight tests where successful compared to the 53% success rate of the HOVER_PID_REF simulations. While the flight tests showed first signs of struggle during the intermediate length transitions, just like the simulations did, the flight tests' probability of success actually got better during the longer transitions.

The greater success of the flight tests compared to the simulations is attributed to the actual aircraft's increased ability to reject yaw disturbances while in hover. Simulation results showed that the aircraft almost always reached the 85 degree pitch angle

PID Control Reference Model Response Flight Tests: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	80%	100%	100%	100%	95%
5	60%	80%	80%	100%	80%
7	100%	80%	100%	40%	80%
10	100%	100%	80%	80%	90%
15	100%	100%	100%	100%	100%
20	100%	100%	100%	80%	95%
Cumulative	93%	95%	95%	88%	93%

Table 7.1: PID Control Reference Model Response Flight Tests: Probability of Success

threshold (where the attitude PID controller gains were increased) but then struggled to keep the yaw error in check when trying to maintain stable hover. This problem with the hover portion of the flight, as opposed to the transition portion, was confirmed by the flight tests results as only 1 out of the 12 failed HOVER_PID_REF flight tests diverged while still in the transitional portion of the test (before the pitch angle exceeded 85 degrees). Just like the simulations, almost all of the HOVER_PID_REF flight test failures occurred when the yaw error diverged, but the actual aircraft was able to correct the yaw errors much more often than not. Additionally, when the flight tests did diverge, they seemed to do so much more slowly than the simulation failures. Figure 7.5 showed that the green results took nearly 12 seconds to go from 5 to 45 degrees yaw error before the aircraft was finally switched to MANUAL mode. In comparison, the simulated failures shown in Figure 6.8 only took between 1 and 5 seconds to experience the same increase in yaw error. The slower divergence of the actual aircraft means that in the event that the hover does fail, the pilot or the autopilot itself, has more time to react and ensure the aircraft stays out of dangerous situations.

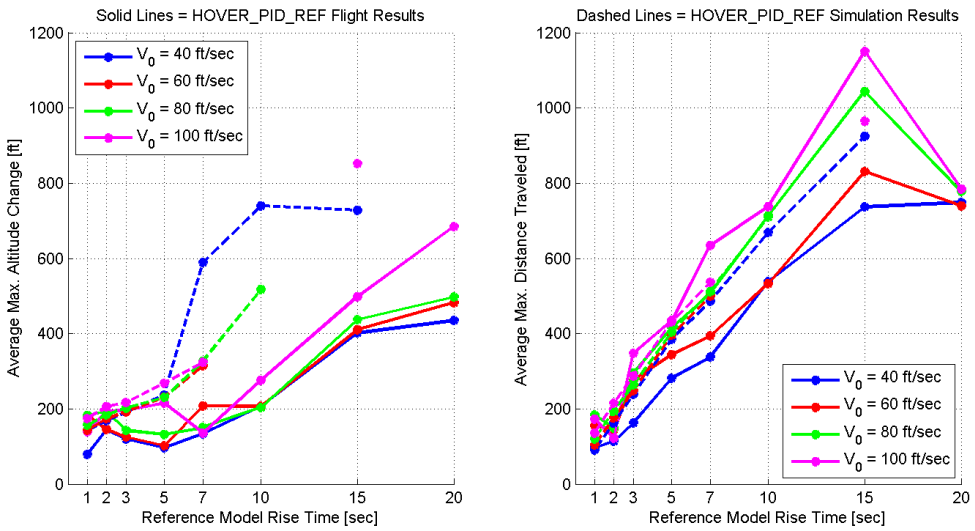


Figure 7.7: PID Control Reference Model Response Flight Test Results: Average HOVER_PID_REF flight test results experience less altitude change than their matching simulations but share the same trends with regards to the distance the aircraft travels.

Figure 7.7 shows the average maximum altitude change and average maximum distance traveled by the aircraft for each of the different HOVER_PID_REF approach speed and rise time combinations. The HOVER_PID_REF simulation results are also shown in Figure 7.7 (as dashed lines instead of solid) for comparison. Trends in the average maximum distance traveled were very similar between the flight tests and simulations. Both the simulation and flight results showed that the distance traveled varied linearly with increases in reference model rise time. Amazingly, the slopes of the linear trends were similar, but the test results showed more variation in the distance traveled when the transition approach speed was changed. This was likely due to the fact that the initial acceleration problem seen in the simulations was not as bad in the flight tests because the actual in-flight elevator PID controller did a better job of trimming the aircraft for level flight at the different speeds than its simulated counterpart did. The only major difference between the simulation and flight test distance results is the large decrease in distance traveled when the reference

model rise time was increased from 15 to 20 seconds. Since none of the simulations were successful with a 20 second rise time, there is no way to tell if the simulations would have resulted in the same decrease, but it is more likely that the change in distance was caused by a major change in wind conditions than a change in controller performance.

The order that the flight tests were completed in was meant to minimize the potential differences in wind conditions between HOVER_PID_REF and HOVER_ADAPTIVE flight tests with the same rise time to help make comparisons between the two controller types. This meant that each of the different reference model rise times for the HOVER_PID_REF flights were tested on a different day. Luckily the average wind speed was about 5 ft/sec for all of the HOVER_PID_REF tests with rise times between 1 and 15 seconds, but the 20 second tests were conducted in winds that averaged over 11 ft/sec. While these tests helped prove that the controllers could handle higher winds, it is likely that the stronger head winds also skewed the distance results by helping the aircraft stop its forward momentum much faster for the 20 second rise time tests than it did for the 15 second tests. Potentially, the linear distance trend would have continued for the 20 second tests if the wind conditions had been the same as they were the majority of the other test days.

Most importantly, Figure 7.7 shows that the theory from Chapter 6, which predicted the average distance traveled plots could be used by the UAV operator to choose a reference model rise time and initial speed to command a transition that ends after a specific distance, was at least partially valid. While the flight test results were much to inconsistent to have any confidence that the aircraft could consistently end the transition-to-hover maneuver within the same 20, 40 or even 80 foot range like the simulations suggested, the flight tests results do suggest that it could be predicted to within 100 or 200 feet which still may be acceptable for some surveillance applications. The flight tests definitely confirmed, however, that actual hover way-

point control would need to be added to the attitude control discussed in this thesis in order to perform any precision maneuvers like perching and vertical landings in crowded areas.

Interestingly, the altitude trends shown in Figure 7.7 don't seem to match at all between the flight tests and simulations except for the very shortest duration transitions. In fact, the trends the flight tests show for rise times between 1 and 10 seconds look much more like the HOVER_ADAPTIVE simulation results seen in Figure 6.17 than the HOVER_PID_REF simulation results. A closer look at the HOVER_PID_REF flight test altitude change is presented in Figure 7.8 and shows some very interesting results.

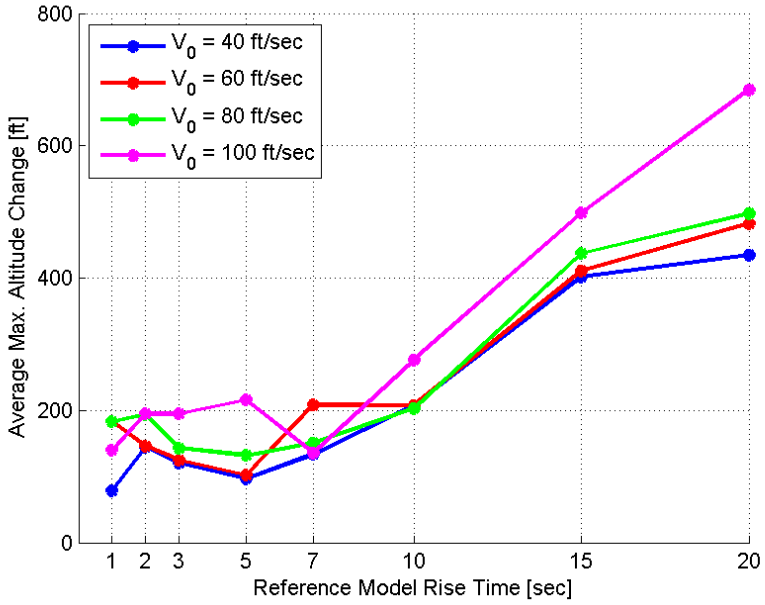


Figure 7.8: PID Control Reference Model Response Flight Test Average Altitude Change

Figure 7.8 shows that for the 40, 60 and 80 ft/sec approach speed tests, the average altitude change actually decreases as the reference model rise time is increased from 2 to 5 seconds. Unlike the HOVER_PID_REF simulations, the actual aircraft seemed to be able to keep the yaw and pitch errors under 5 degrees during the transition

more effectively than the simulated aircraft could. This meant that there was less kinetic energy added to the system since the hover divergence throttle did not have to be engaged for as much time in the flight tests as it did in the simulations. Since the aircraft spends more time at low pitch angles the longer the reference model rise time is, more kinetic energy could be bled off due to drag during the intermediate transitions instead of being converted into potential energy through altitude increase. The altitude change starts to increase again after the rise time exceeds 7 seconds, because the aircraft starts to spend more and more time at very high pitch angles where the controllers have to work very hard to track the reference model before finally reaching 85 degrees pitch where the PID gains increase to their hover values. The extra work the PID controllers have to do for the longer transitions ends up translating into increased need for the divergence throttle which then results in more altitude gain as the aircraft struggles to maintain the desired attitude.

While there is no real trend in the average altitude change for reference model rise times below 10 seconds, all of the results are at least consistently between 100 and 200 feet so the UAV operator at least has some idea of what the end altitude of the transition maneuver might be. The altitude appears to transition to a linear relationship once the rise time exceeds 10 seconds, but with only 2 rise times longer than that, it's hard to make a concrete conclusion. More tests would need to be conducted to be confident that the altitude change becomes linear for longer real flight transitions.

The HOVER_PID_REF flight test proved the attitude PID controllers are capable of controlling both short and long duration transition-to-hover maneuvers of a fixed-wing aircraft. For short transitions, the aircraft carries enough kinetic energy through the transitional portion of the maneuver so that the pitch response is smooth and tracks the reference model fairly closely. During longer transitions, the aircraft eventually bleeds off its kinetic energy through drag and the elevator and throttle control

have to work together to help the aircraft maintain altitude while slowly increasing the pitch angle. Unfortunately, flight tests results showed that simply changing the reference model rise time is not a predictable way of controlling the altitude and downrange distance change of the aircraft during the transition as the simulation results had suggested.

In general, the HOVER_PID_REF flight mode proved to be an easy way to program a robust and effective transition-to-hover controller for the ArduPilot autopilot. The HOVER_PID_REF flight tests were much more successful then the matching simulations as they were able to transition to and maintain stable hover for 93% of the attempted tests. It was also relatively easy to tune the controller in its final state since the user only needs to establish an initial set of level flight gains and then determine the appropriate hover speed scalar that is needed to maintain hover flight instead of having to tune two full sets of PID gains. While changing the reference model rise time did not end up being a reliable way of controlling the aircraft's position at the end of the transition, changing the rise time was an effective way of making the transition smoother for a specific airframe. By tuning the rise time, the UAV operator can ensure that the aircraft reaches 90 degrees pitch angle at the same time as it runs out of kinetic energy and stops climbing. Doing this results in a very smooth transition for the aircraft that has minimal altitude increase and puts minimal stress on the airframe, while giving the controller the best chance of maintaining stable hover by providing a very stable starting point. Independent of the HOVER_ADAPTIVE flight tests results, the HOVER_PID_REF results served as proof that small amounts of user control can be added to improve the transition maneuver while still maintaining its ease of use and compatibility with the ArduPilot autopilot system when compared to the HOVER_PID_STEP flight mode.

7.3 Model Reference Adaptive Control

As discussed in Chapters 1 and 4, the purpose of incorporating Model Reference Adaptive Control into the ArduPilot's newly created flight modes was to hopefully improved the transition to hover performance of the autopilot while still maintaining, or even improving, the ease of use of the system. Unfortunately, initial simulations and flight tests showed that the MRAC algorithm was not well suited for maintaining stable hover, so a combination of both MRAC and PID controllers was used to control the aircraft's attitude when in the HOVER_ADAPTIVE flight mode. A large increase in the complexity of the system compared to the HOVER_PID_REF flight mode was avoided, however, by setting the initial values of the adaptive gain matrix equal to the proportional gains from the level flight attitude PID controllers (reasons for this were presented in Chapter 4), which meant that the only additional values the UAV operator has to determine for the HOVER_ADAPTIVE mode, compared to the HOVER_PID_REF mode, are the values of the adaptive parameter matrix in the MRAC algorithm. So while the ease of use of the hover flight modes was not improved by incorporating MRAC, it was not completely ruined by the addition, and the use of MRAC still had the potential to be worth while.

The usefulness of the HOVER_ADAPTIVE flight mode would then be largely dependent on its success rate and level of precision relative to the HOVER_PID_REF mode. Simulation results presented in Chapter 6 showed HOVER_PID_REF and HOVER_ADAPTIVE flight modes were both expected to successfully transition to and maintain stable hover for about 55% of the attempted tests, but that the successful HOVER_ADAPTIVE simulations typically tracked the chosen reference model more closely and experienced less altitude gain than the analogous HOVER_PID_REF simulations. Flight test results presented in Section 7.2, however, showed that HOVER_PID_REF flights were actually successful 93% of the time so it will be difficult for

the HOVER_ADAPTIVE flight mode to improve on that level of reliability. It is still possible that the potential increased level of precision of the HOVER_ADAPTIVE flight mode may make it very useful for some applications even if it cannot improve the level of reliability compared to the HOVER_PID_REF mode.

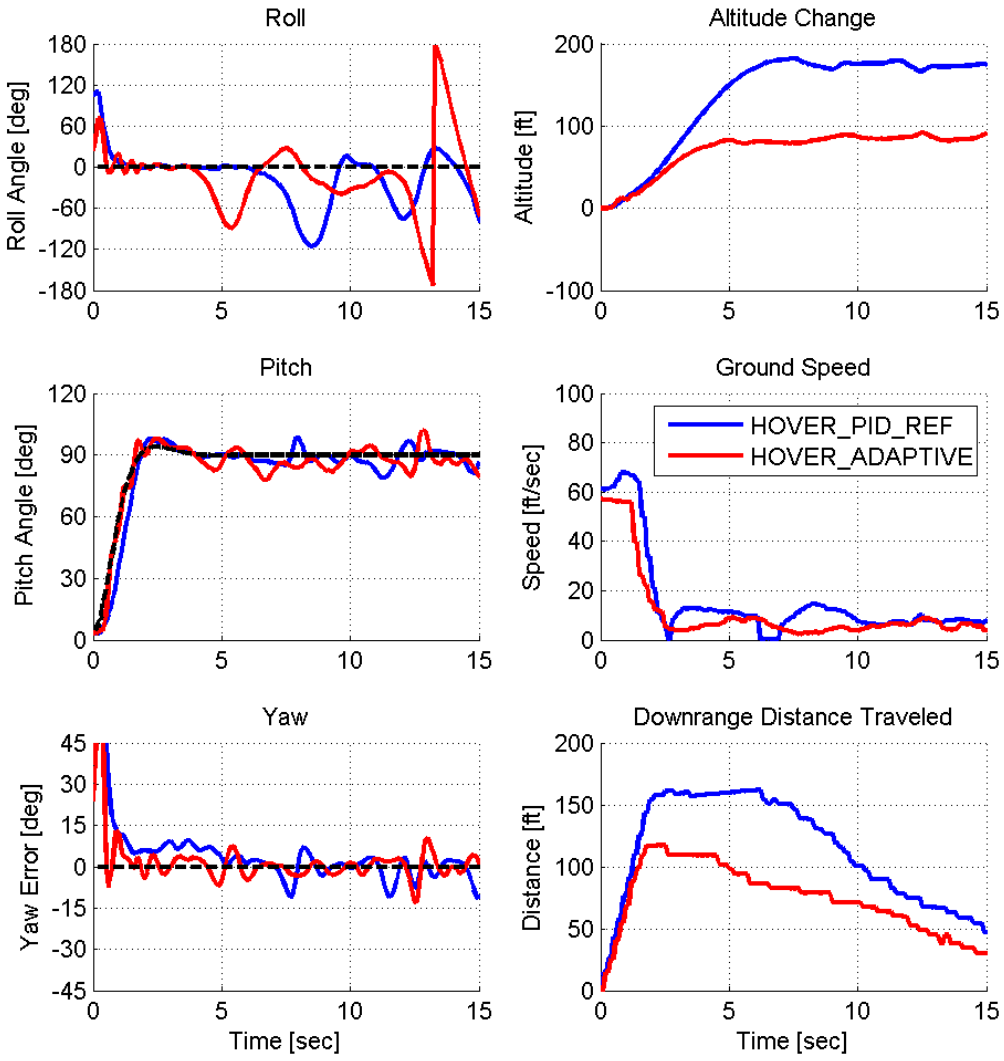


Figure 7.9: MRAC Flight Test Example: HOVER_ADAPTIVE flight results with approach peed = 60 ft/sec and rise time = 1 sec, compared to HOVER_PID_REF flight results with matching conditions.

Figure 7.9 shows an example of one of the HOVER_ADAPTIVE flight test re-

sults compared to a HOVER_PID_REF flight test with matching conditions. The results shown in Figure 7.9 back up those that were seen in simulations where the MRAC was able to track the reference models better than the attitude PID controllers during the transitional portions of the flight. The pitch response plots show that the HOVER_ADAPTIVE controlled aircraft actually oscillates around the pitch angle reference model (shown as a black dashed line) during the transition while the HOVER_PID_REF controlled aircraft's pitch angle lags behind desired pitch for most of the transition. The oscillations caused by the MRAC are very small (typically less than 2 degrees) and were barely noticeable to the naked eye during flight test, but the increased ability to track the reference model during the transition eliminated the need for intervention by the divergence throttle logic which meant the HOVER_ADAPTIVE controlled aircraft climbed nearly 100 feet less during the test than the HOVER_PID_REF controlled aircraft did.

The increased tracking ability of the MRAC continued even as the reference model rise time was increased, as demonstrated in Figure **??**. Each of the four flight test results (Rise time: Blue = 2 sec, Red = 3 sec, Green = 5 sec, Magenta = 7 sec) are plotted against their respective reference models which are shown as black dashed lines for contrast. The differences between the reference model lines and the actual aircraft pitch angles are nearly indistinguishable during the transitional portion of the flight except for the same small MRAC induced oscillations that were seen in Figure 7.9. It is not until the pitch angle exceeds 85 degrees and the attitude PID controllers take over control of the airplane that the roll, pitch, and yaw errors start vary more intensely.

Figure 7.11 provide further evidence that the HOVER_ADAPTIVE flight test results were very similar to the HOVER_PID_REF flight results except for the decreased use of the divergence throttle. The average distance trends are almost identical between the HOVER_ADAPTIVE and HOVER_PID_REF flight tests results, but there

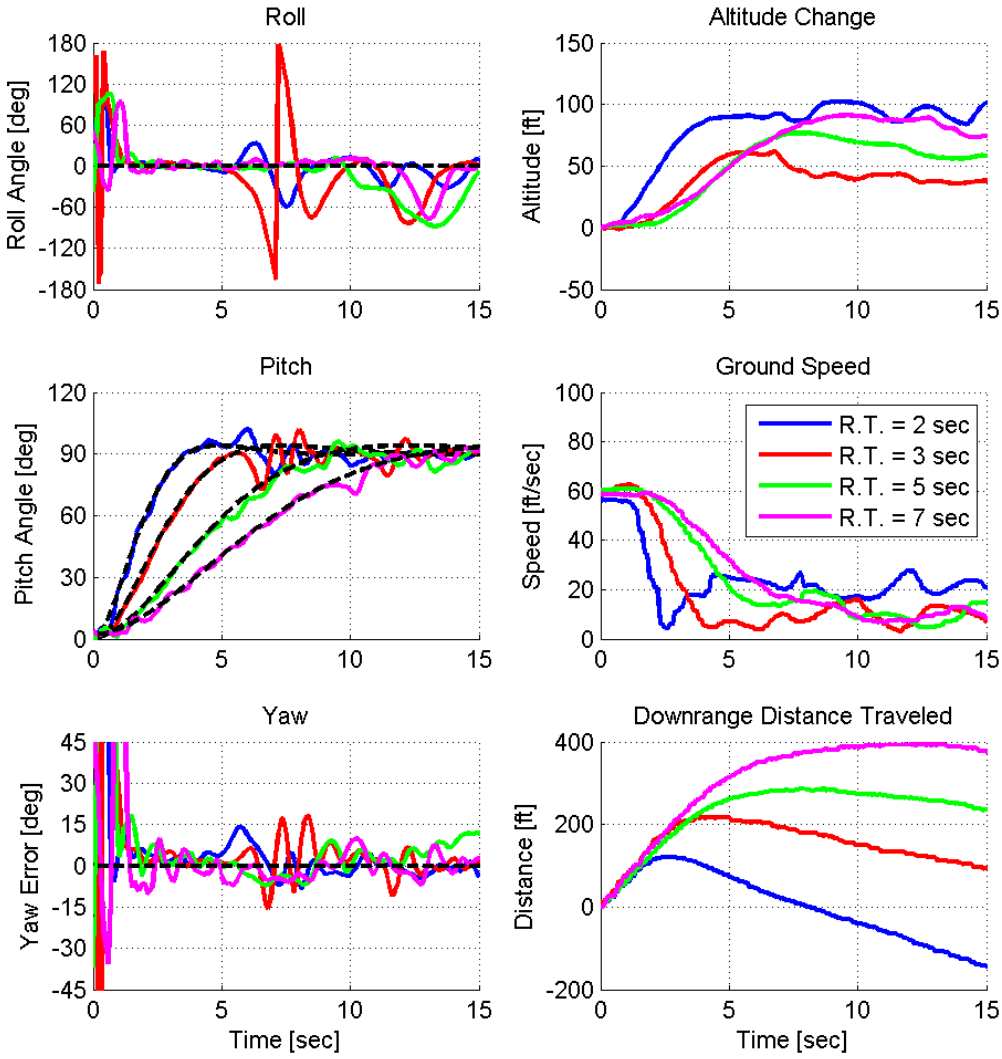


Figure 7.10: MRAC Flight Test Precision Tracking

are small differences in the average altitude gain trends. For reference model rise times between 1 and 10 seconds, the altitude change trends for the two flight modes are fairly constant but the HOVER_ADAPTIVE flight mode, in general, experiences slightly less altitude gain. The results of the 15 second rise time tests show that the MRAC transitions had less altitude gain than the same PID controlled transitions but there is no way to tell if that trend would have continued since none of the 20

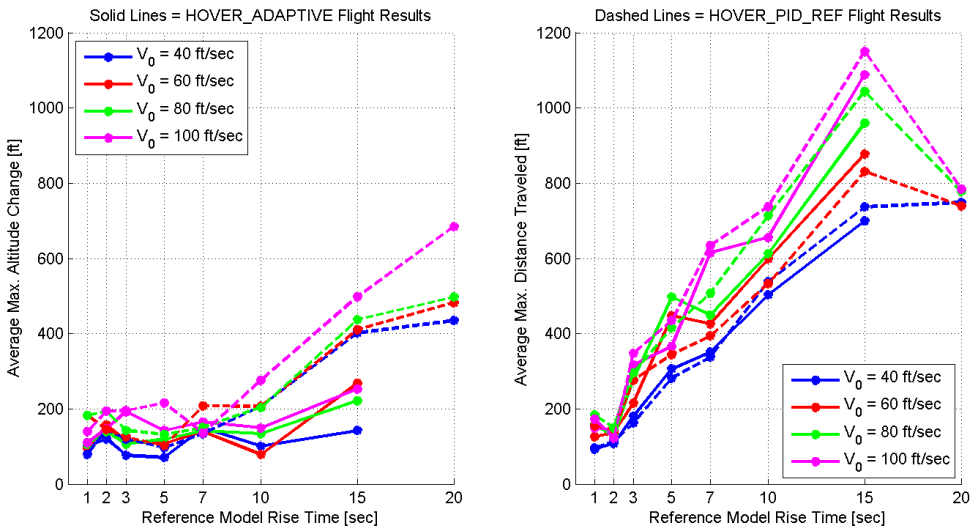


Figure 7.11: MRAC Flight Test Results: HOVER_ADAPTIVE average distance traveled trends matched those of the HOVER_PID_REF flights almost exactly, but the HOVER_ADAPTIVE average altitude change was slightly less for most flight conditions.

second tests were successful for the HOVER_ADAPTIVE flight mode.

The biggest difference between the HOVER_ADAPTIVE and HOVER_PID_REF flight tests was that the MRAC transitions had much lower success rates than their PID controlled counterparts did. Table 7.2 shows that 71% of the attempted HOVER_ADAPTIVE flight tests were successful. While this is an improvement from the 56% of HOVER_ADAPTIVE simulations that were successful, it is quite a bit lower than the 93% success rate of the HOVER_PID_REF flight tests.

Unlike the simulations however, most of the HOVER_ADAPTIVE flight tests failures occurred while the MRAC algorithm was still in control of the airplane and not after the attitude PID controllers had taken over. In fact, only 4 of 47 total unsuccessful HOVER_ADAPTIVE fight tests failed after the PID controllers had taken control of the aircraft and all 4 of those failures occurred almost immediately after the PID controllers took over, instead of after the aircraft had already been hovering for a few seconds like the was so common in the simulations. An example

MRAC Flight Tests: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
5	100%	80%	60%	100%	85%
7	100%	100%	80%	80%	90%
10	100%	60%	20%	40%	55%
15	40%	40%	20%	40%	35%
20	0%	0%	0%	0%	0%
Cumulative	80%	73%	60%	70%	71%

Table 7.2: MRAC Flight Tests: Probability of Success

of this is shown in Figure 7.12. During the longer transitions, the values in the adaptive gain matrix grow because of the small roll, pitch, and yaw errors that occur throughout the transition due to wind and controller overshoot. By the time aircraft gets close to the end of the transition, even small wind disturbances have the potential to cause an overreaction by the now very high gain system. Figure 7.12 shows that the controller started to struggle to maintain the desired yaw and pitch angles of the aircraft just before the 15 second mark of the flight test. Both the rudder and elevator then overreact due to the high gain of the system at that point, which causes the pitch angle to jump to nearly 120 degrees and the yaw error to exceed 45 degrees in less than one second. As soon as the pitch angle increases past 85 degrees, control of the airplane was handed over to the attitude PID controllers but at that point the yaw error was already too large to be recovered and the aircraft continued to diverge before the pilot finally takes control of the airplane.

While MRAC failures like the one shown in Figure 7.12 are understandable because they occur when the airplane is at very high pitch angles and very low speeds, they only account for 9% of all the HOVER_ADAPTIVE flight test failures. Another

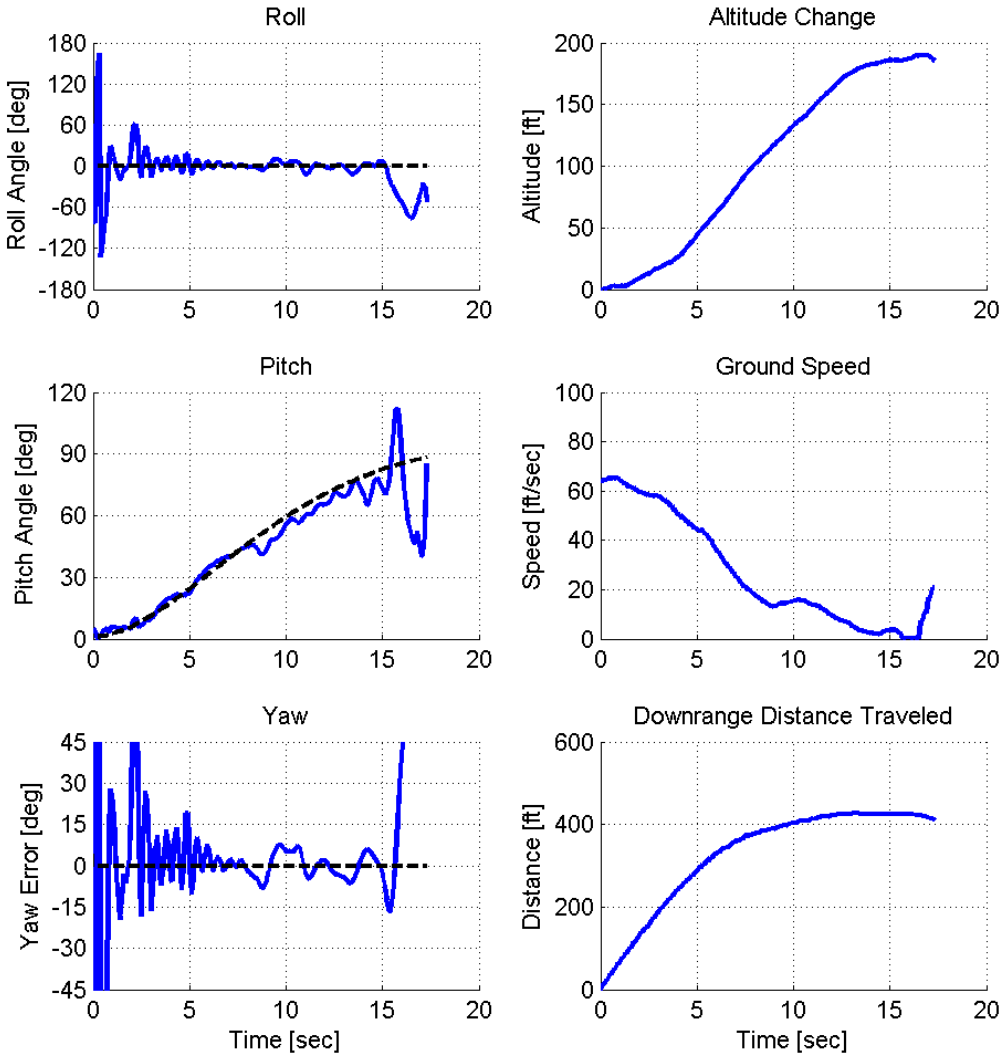


Figure 7.12: MRAC Flight Test Failure After PID Controllers Take Over

3 out of the 47 failures also occurred while the airplane was at relatively high pitch angles but never got far enough into the transition for the PID controllers to have a chance to stabilize the airplane before diverging. The other 40 failures, however, occurred before the aircraft ever reached a pitch angle of 60 degrees. Figure 7.13 shows a very good example of how the MRAC transitions struggled to control the aircraft for more than a few seconds when at low pitch angles and high speeds.

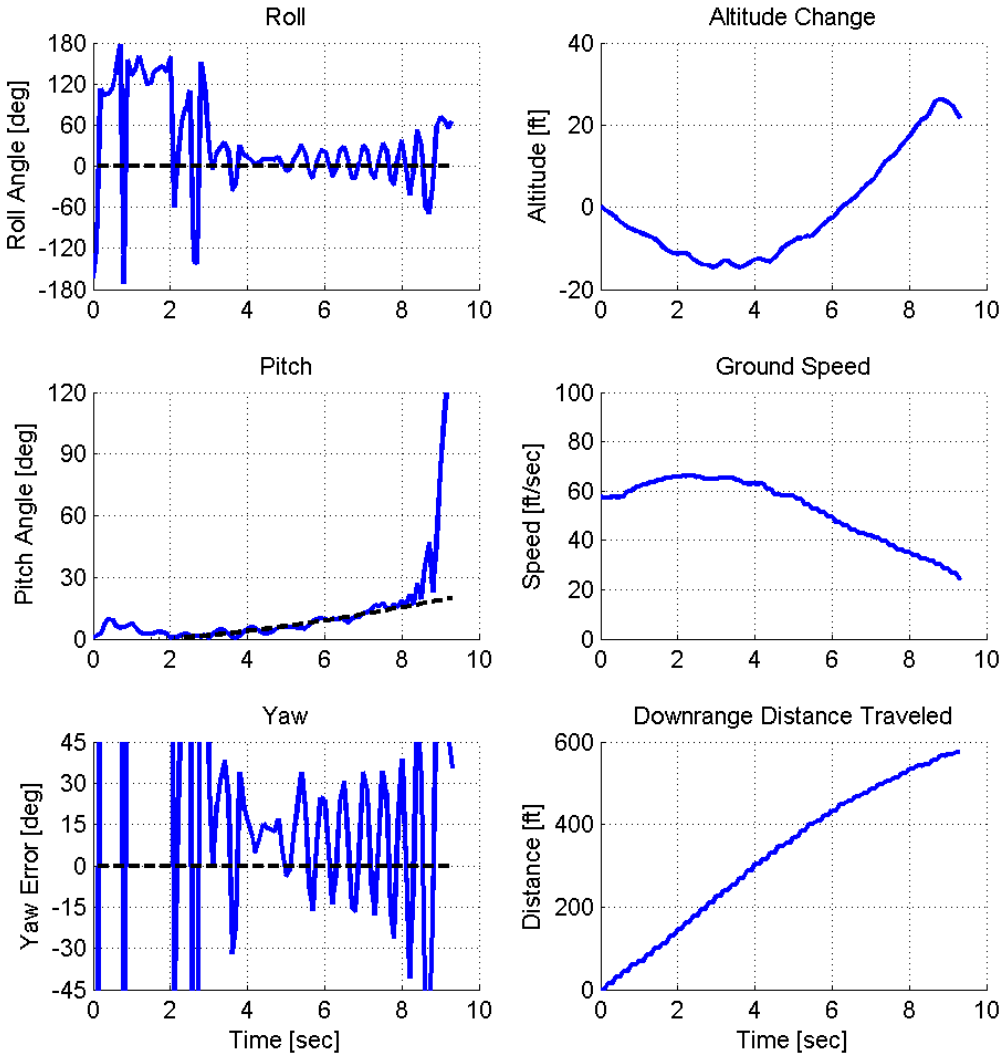


Figure 7.13: MRAC Flight Test High Speed, Low Pitch Angle Failures

The HOVER_ADAPTIVE flight test shown in Figure 7.13 experiences the same sensor "dizziness" that was discussed in Section 7.2 for the first 4 seconds of the flight. After the "dizziness" dissipates the roll, pitch and yaw errors are all relatively small, but then the yaw and roll errors start to violently oscillate until the errors become so large that the aircraft becomes completely unstable and has to be returned to manual control. Similar roll and yaw oscillations can be seen in the early seconds of Figure

7.12, but those oscillations diminish and eventually stop as time goes on instead of continuing to grow like the ones in Figure 7.13.

These unstable yaw and roll oscillation are likely caused by the same adaptive gain growth problem that made it necessary to switch to PID control once the aircraft reached hover orientation. When the aircraft is at low pitch angles and high speeds, the gain margin for the yaw and roll axes is much smaller than it is when the aircraft is at high pitch angles and low speeds. As the adaptive controller tries to eliminate the initial roll and yaw errors while the aircraft is still close to level flight, the initial values of the adaptive gain matrix are high enough to cause small amounts of overshoot in the controller response. For the shorter duration transitions, the aircraft starts to pitch up and slow down more quickly so the gain margin grows at a faster rate than the gain increase from the adaptive algorithm does, and the overshoot oscillations eventually dissipate. For longer duration transitions, however, the aircraft stays at high speeds and low pitch angles for a longer period of time so the initial gain margin remains unchanged for longer as well. In many of those cases, the adaptive gains grow faster than the gain margin which means the gain margin is eventually exhausted. This then causes the initial overshoot oscillations to grow instead of dissipating since the controller response is now out of phase with the actual roll and yaw dynamics of the aircraft. The adaptive gains then grow at an even faster rate since the yaw and roll errors continue to grow until the aircraft becomes completely unstable.

The runaway adaptive gain growth problem can be solved however, by decreasing the magnitude of the roll and yaw terms in the adaptive parameter matrix and the initial values of the adaptive gain matrix. Decreasing the initial values of the adaptive gain matrix makes the controller act like a lower gain system to start with and then decreasing the adaptive parameter matrix terms also causes the roll and yaw gains to grow more slowly as time goes on, which together help alleviate the adaptive gain growth problem. For example, if the roll and yaw terms in both of the adaptive

matrices are cut in half, compared to their original values, so that

$$G_{e_0} = \begin{bmatrix} k_{p_{roll}}/2 & 0 & 0 \\ 0 & k_{p_{pitch}} & 0 \\ 0 & 0 & k_{p_{yaw}}/2 \end{bmatrix} = \begin{bmatrix} 0.075 & 0 & 0 \\ 0 & 0.27 & 0 \\ 0 & 0 & 0.175 \end{bmatrix}, \tag{7.1}$$

and

$$H = \begin{bmatrix} 0.005 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.005 \end{bmatrix}, \tag{7.2}$$

then longer duration transitions can be completed successfully like the flight test shown in Figure 7.14.

Figure 7.14 shows that adjusting the adaptive matrices to the above values enables the HOVER_ADAPTIVE flight mode to successfully complete one of the 20 second rise time flight tests, which had a 0% success rate with the original gain values. Decreasing the initial values of the adaptive gain matrix helps minimize the initial overshoot of the controllers while the aircraft is still at high speed, and since the transition is so long, the lower values in the adaptive parameter matrix cause the adaptive gain matrix values to grow more slowly so that they are at the appropriate levels when the aircraft really needs the extra gain at low speeds and high pitch angles. Some controller induced yaw oscillations still occurred during the flight test shown in Figure 7.14, but not until 25 seconds had already past. At this point the aircraft had already exceeded 60 degrees pitch angle and was at very low speed so the controller was able to reject the wind disturbance that caused the initial error without much difficulty. The MRAC was then able to successfully hand off control of the aircraft to the attitude PID controllers and stable hover flight was maintained for the remainder of the test. (Note: The constant altitude increase during the last 60 seconds of the test was caused by the pilot accidentally moving the throttle stick on the transmitter so that a steady climb rate of about 1.2 ft/sec was commanded. These results show

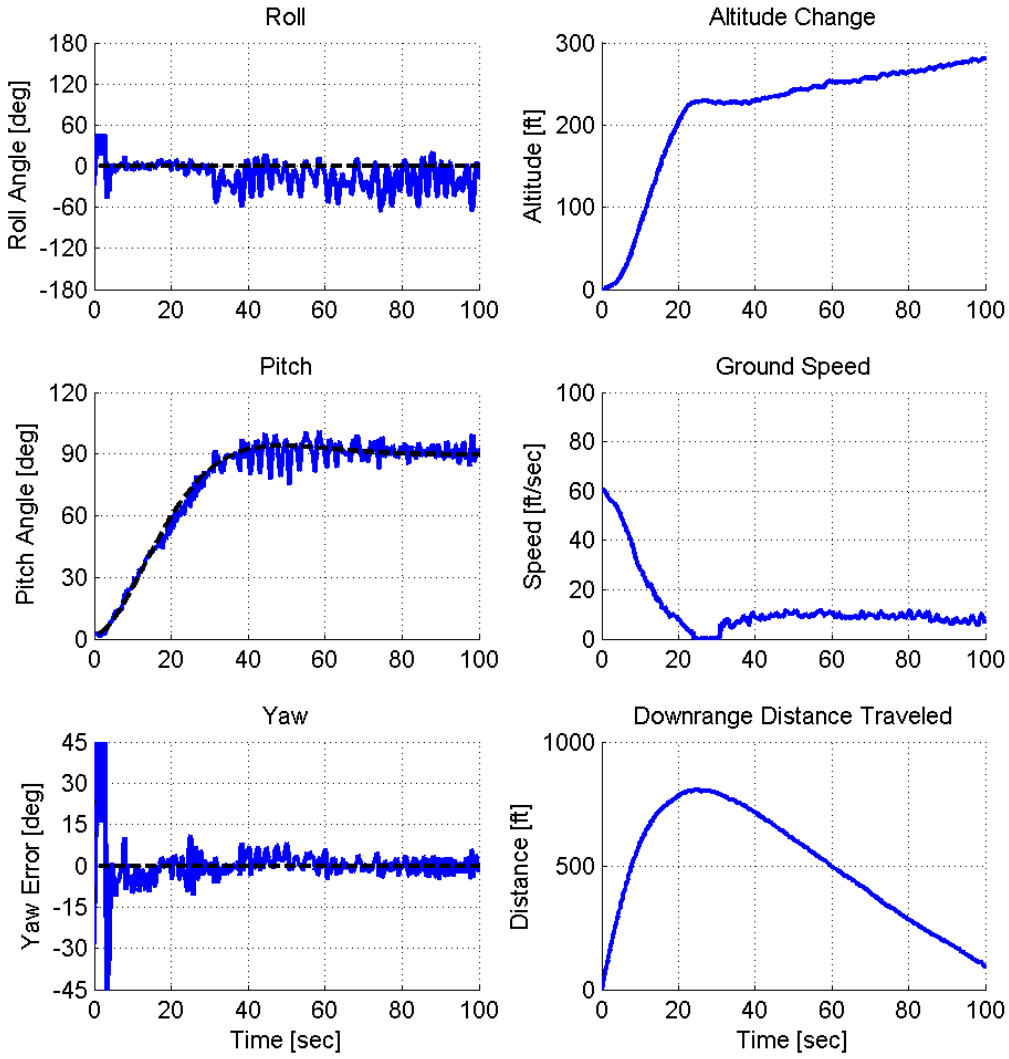


Figure 7.14: MRAC Flight Test With Adjusted Gains

that the throttle controller was able to hold non-zero climb/descent rates as well.)

While changing the adaptive gains like this can increase the probability of success during longer transitions, doing so may affect the results of the HOVER_ADAPTIVE mode for shorter reference model rise times. The initial precision of shorter MRAC transitions may be hurt by decreasing the adaptive gains, so more responsibility is put on the UAV operator to choose gains that are appropriate for the

application. This does not mean that development of the HOVER_ADAPTIVE flight mode was unsuccessful, or that it is not a useful flight mode. It simply means that the the HOVER_ADAPTIVE mode is slightly less "plug-and-play" like, when compared to the HOVER_PID_STEP, HOVER_PID_REF, and standard ArduPlane level flight modes.

Overall, using Model Reference Adaptive Control, instead of PID controllers, to control the transition-to-hover of a fixed-wing UAV had both positive and negative effects. For shorter duration transitions, the MRAC algorithm helped the aircraft track the desired reference model more closely during the transitional portion of the flight and indirectly helped decrease the average altitude gained during the transitions compared to PID controlled transitions with matching conditions. The HOVER_ADAPTIVE flight tests also showed very similar success rates to the HOVER_PID_REF flights when the reference model rise time was 7 seconds or less, but struggled heavily during longer transitions because of the runaway adaptive gain growth problem.

The HOVER_ADAPTIVE flight test results did prove that Model Reference Adaptive Control is a viable solution for controlling a transition-to-hover maneuver, but it was not as robust and easy to use as originally intended. While the MRAC transitions are more precise, a single MRAC design cannot handle the same variety of input reference models compared to the PID controllers. Based on this fact, it can be said that Model Reference Adaptive Control should only be used for specific transition applications where increased precision is needed, while the basic attitude PID controllers are a better fit for the ArduPilot system as a whole, since it is designed to be used in a variety of different airframes and be simple to tune even for someone not trained in advanced control theory.

7.4 Hover-to-Level Transition

Hover-to-level transition flight tests were intended to validate the results from the hover-to-level simulations that showed the autopilot should be able to safely and efficiently transition the aircraft from hover back to level flight by simply switching from one of the three hover flight modes into any of the standard ArduPlane level flight modes. The flight tests were conducted in the same way as the hover-to-level simulations in that the autopilot was put in one of the three hover flight modes and allowed to maintain stable hover for a short period of time before being switched into the STABILIZE flight mode. Once in STABILIZE, the pilot would immediately set the throttle to 75% and then leave it there for the duration of the transition back to level flight.

During the hover-to-level flight testing, it was discovered that a small, but important, detail was missed when setting up the matching simulations. When the autopilot is switched between any of the level flight modes, the integrator terms of the attitude PID controllers are not reset in order to maintain the trimmed state of the aircraft. When the firmware was modified to add the three new hover flight modes, the integrator terms were mistakenly allowed to carry over from the hover flight modes to the level flight modes in same way that they transfer between the level modes. This presented a problem when switching the aircraft from a hover flight mode into STABILIZE for the hover-to-level flight tests, because the aileron PID controller often experiences large amounts of integrator windup during hover due to its attempts to counteract the motor torque. When the aircraft was switched to STABILIZE in the hover-to-level flight tests, the excess aileron integrator term took a few seconds to dissipate and actually caused the aircraft to circle around its initial location instead of flying away at constant heading. Since the rudder is not used in STABILIZE mode, the aircraft kept circling until the steady state roll error was eventually eliminated at

which point the aircraft held the constant heading, steady level flight path that was originally intended. An example of this is shown in Figure 7.15.

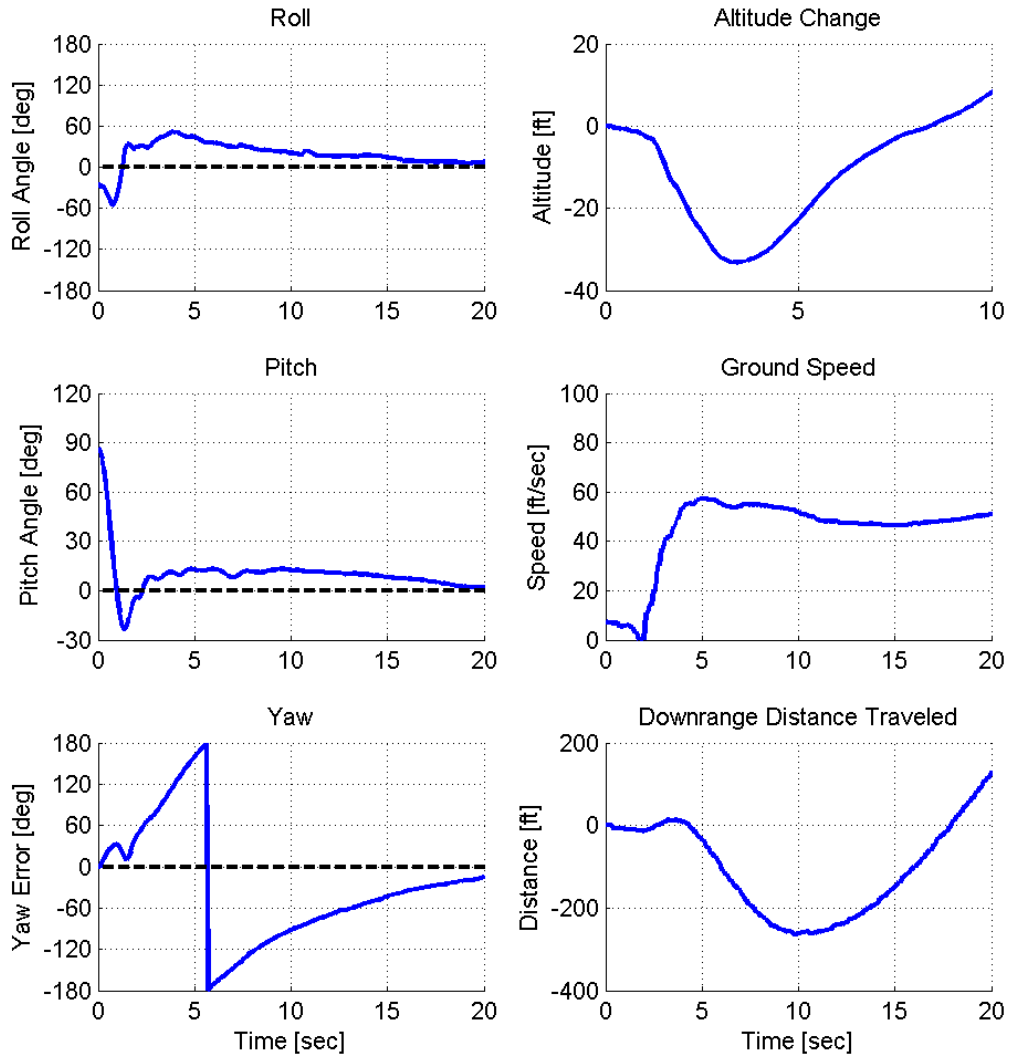


Figure 7.15: Hover-to-Level Flight Test Example

Figure 7.15 shows that after the airplane initially returns to level orientation and begins to build airspeed, the roll angle jumps to nearly 60 degrees due to the large aileron integrator value which was carried over from hover flight. The yaw error shows that the aircraft actually completed one full circle before the steady state roll error

was eliminated and the aircraft stopped turning. The fact that the final heading was nearly identical to the original heading was pure coincidence.

Luckily, the pitch response of the test aircraft was very similar to that of the simulated aircraft shown in Figure 6.19 in that there is about 15 degrees of initial pitch angle overshoot before the airplane returns to a small positive pitch angle. The pitch error then slowly dissipates as the altitude and airspeed stabilize.

The fact that the pitch response of the aircraft matched that of the simulations so closely meant that simply resetting the PID controllers' integrator terms when the autopilot was switched from a hover flight mode to a standard level mode would undoubtedly fix the circling problem and produce hover-to-level transition flight test results that were very similar to the simulations. Unfortunately, the Yak 54 model that was used to conduct all of the flight tests in this thesis suffered a fatal crash before the integrator problem could be fixed and the hover-to-level transitions could be retested. Only two of the faulty tests were able to be completed before the crash occurred but both showed the same ability of the autopilot to safely and efficiently control the pitch angle of the aircraft despite the circling problem. It was decided that the test aircraft would not be rebuilt as all of the other necessary tests had already been completed. The faulty flight tests were enough to prove the concept that the autopilot could transition the aircraft from hover back to level flight using the standard ArduPlane level flight modes and only required the very minimal firmware modification that the attitude PID controller integrator terms be reset when switching back to one of the level flight modes.

8 CONCLUSION

This work was intended to show that autonomous transition-to-hover of fixed-wing UAVs could be achieved using the low-cost, open source ArduPilot-Mega autopilot. Three new hover flight modes were developed for the autopilot, each with a different control logic for transitioning to and maintaining stable hover, but all intended to maintain the ease of use that the APM system is known for. First, the HOVER_PID_STEP flight mode was designed to use the APMs standard attitude PID controllers to track a step response where the desired pitch angle was instantaneously increase to 90 degrees. Next the HOVER_PID_REF flight mode used the same attitude PID controllers but tried to track a user defined second order reference model which was designed to effectively lengthen or shorten the duration of the transition maneuver. Finally, the HOVER_ADAPTIVE flight mode used Model Reference Adaptive Control to control the aircraft's attitude during the transitional portion of the flight in an attempt to increase performance, but then handed control back to the attitude PID controllers once stable hover was achieved. All three flight modes took advantage of the quaternion based Euler angle error method, described in Chapter 3, to eliminate the risk of gimbal lock and ensure the aircraft reacts properly when in hover orientation.

A Matlab and Simulink simulation was created and proved to be a useful tool in development of the hover flight modes' controller logic. While deficiencies in the calculations of the aerodynamic rudder forces prevented the simulated aircraft from maintaining stable hover for long periods of time, the simulation results showed very similar trends for the transitional portions of the flight when compared to actual flight test results. The simulations also helped discover that the Model Reference Adaptive

Control algorithm used in the HOVER_ADAPTIVE flight mode was poorly suited to hold the aircraft in hover orientation for more than a few seconds, even though it was good at transitioning the aircraft from level to hover flight.

Once the designs of the transition-to-hover controllers were finalized, a series of flight tests were conducted to observe the performance of each of the three flight modes under various conditions. Flight tests results showed that the HOVER_PID_STEP flight mode was a very reliable method for the transition maneuver as all 20 of the attempted tests were successful. The initial severity of the step response did, however, put a lot of stress on the airframe which might be undesirable for applications with sensitive payloads. The addition of the second order reference model for the control input of the HOVER_PID_REF flight mode proved to be very successful. The rise time of the desired pitch response was varied from 1 to 20 seconds to test the versatility of the attitude PID controllers which still managed to complete 93% of the attempted tests. By allowing the user to change the design of the reference model, he or she can easily command a transition maneuver that is appropriate for any airframe and payload combination. Shorter duration transitions resulted in less altitude change and distance traveled by the aircraft, but longer transitions put less strain on the aircraft. Flight test results of the HOVER_ADAPTIVE flight mode showed that the Model Reference Adaptive Control based transitions had increased precision in the tracking of the input reference model during the transition, but were also less reliable since only 71% of the attempted tests were successfully completed. Further investigation showed that the most of the failures occurred during the longest duration transitions and were caused by excessive adaptive gain growth when the aircraft spent large periods of time at low pitch angles and high speeds. MRAC still proved to be an effective way of controlling the transition-to-hover maneuver but just required small tweaks to the adaptive gains to maintain the same level of reliability when the desired length of the transition was increased.

Most importantly, the newly developed flight modes managed to maintain the same ease of use that makes the ArduPilot system popular for both researchers and hobbyists. Use of the HOVER_PID_STEP and HOVER_PID_REF flight modes only require that the UAV operator tunes a single set of PID gains while the aircraft is safely in level flight, and then determines the appropriate hover speed scalar value that is needed to maintain stable hover. The HOVER_ADAPTIVE flight mode uses the same level flight PID gains to populate the initial conditions of the adaptive gain matrix and then only requires the operator to choose the values in the adaptive parameter matrix, which are relatively easy to tune even for a novice. All of the gain tuning needed to successfully use any of the three hover flight modes can be accomplished by simple trial and error during flight tests and does not require any advanced knowledge of control theory to achieve satisfactory transition-to-hover performance. In addition, it was shown that the addition of the hover flight modes did not affect the autopilot's ability to function in normal level flight as flight tests results showed that the autopilot could safely and efficiently transition the aircraft from hover back to level flight by simply switching into any of the standard ArduPlane level flight modes. This level of accessibility was one of the main goals of this project and the final version of the firmware, with the newly created hover flight modes, was even shared with the DIY Drones developer and hobbyist community in the hope that users around the world would try the flight modes in their own aircraft and continue to make additions and improvements.

8.1 Future Works

It would be interesting for the flight tests conducted as part of this thesis to be repeated using an airframe that was specifically designed for VTOL flight. While the Carbon-Z Yak 54 model used in this thesis had plenty of power, the single engine configuration meant that torque roll was a major problem while the aircraft was

hovering. Using an airframe with two engines that rotated in opposite directions, or one engine with two contra-rotating propellers, would eliminate most of the motor torque issues and allow for more precise pointing of the aircraft while hovering.

Expanding the capabilities of the ArduPilot to include fly-by-wire style and fully autonomous GPS based position control of the aircraft while it is hovering would also be an interesting addition to the work done in this thesis. By using the transition-to-hover flight modes developed in thesis for the inner loop attitude stabilization, one could develop outer loop controllers that determined the desired attitude of aircraft so that controlled strafing could be achieved during hover. This inner and outer loop controller configuration is the same way the ArduPilot controls the aircraft when in fully autonomous or fly-by-wire level flight modes, so it is feasible that it would work for controlling hover flight as well. Development of these controllers would need to be done in an airframe specifically designed for hover flight, though, since the torque roll problems mentioned above would make controlled lateral motion while hovering very difficult.

Finally, attempting to develop a new formulation of adaptive control that does not experience the same adaptive gain runaway problem that the Model Reference Adaptive Control algorithm did in this thesis would be a very worthwhile endeavor. While the results of this thesis did show that adaptive control has the potential to improve transition-to-hover performance by increasing the gain in the system as the plant changes throughout the transition, the MRAC formulation used was not well suited to continue controlling the aircraft once in stable hover. Developing a new type of adaptive control that capped the gain growth, or possibly even had a way of reducing the adaptive gains once the aircraft made it through the most difficult parts of the transitions, may eliminate the need to switch to the attitude PID controllers at the end of the transition maneuver.

Bibliography

[1] Ira Abbott and Albert Von Doenhoff. *Theory of Wing Sections*. Dover Publications, New York, 1959.

[2] AeroVironment, Inc. UAS Advanced Development: SkyTote. Website, 2013. http://www.avinc.com/uas/adc/skytote/.

[3] AeroVironment, Inc. UAS: RQ-11B Raven. Website, 2013. https://www.avinc.com/uas/small_uas/raven/.

[4] Gavin Ananda. UIUC Propeller Database. Online database, University of Illinois at Urbana-Champaign, 2011. http://www.ae.illinois.edu/m-selig/props/propDB.html.

[5] John Anderson, Jr. *Fundamentals of Aerodynamics*. McGraw Hill, New York, NY, fifth edition, 2011.

[6] Arduino Team. Arduino. Website, May 2013. Home website for Arduino Project: http://arduino.cc/en/.

[7] Pierre Richard Bilodeau, Eric Poulin, Eric Gagnon, Franklin Wong, and Andr Desbiens. Control of a hovering mini fixed wing aerial vehicle. AIAA Guidance, Navigation, and Control Conference Paper AIAA 2009-5794, Defence Research and Development Canada Valcartier, Qubec City, Qubec, August 2009. http://arc.aiaa.org/doi/abs/10.2514/6.2009-5794.

[8] Biometrics & Dexterous Manipulation Laboratory. Perching home. Website, May 2013. http://bdml.stanford.edu/Main/PerchingHome.

[9] John Brandt and Michael Selig. Propeller performance data at low Reynolds numbers. Conference Paper AIAA 2011-1255, University of Illinois at Urbana-

Champaign, January 2011. http://www.ae.illinois.edu/m-selig/pubs/BrandtSelig-2011-AIAA-2011-1255-LRN-Propellers.pdf.

[10] Chris Critzos, Harry Heyson, and Robert Boswinkle. Aerodynamic characteristics of NACA 0012 airfoil section at angles of attack from 0 to 180 degrees. Technical note 3361, Langley Aeronautical Laboratory, Langley, VA, January 1955.

[11] DIY Drones Community. DIY Drones. Website, July 2013. Website is home to forums for ArduPilot users and developers: http://diydrones.com/.

[12] DIY Drones Development Team. *ArduPilot-Mega: Official ArduPlane Repository*. DIY Drones, July 2013. https://code.google.com/p/ardupilot-mega/wiki/home?tm=6.

[13] E-flite. Carbon-Z Yak 54. Website, 2013. Manufacturer's information page: http://www.e-fliterc.com/Products/Default.aspx?ProdID=EFL10075.

[14] Adrian Frank, James McGrew, Mario Valenti, Daniel Levine, and Jonathan How. Hover, transition, and level flight control design for a single-propeller indoor airplane. Technical report, Massachusetts Institute of Technology, Cambridge,MA, May 2007. https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CC8QFjAA&url=http%3A%2F%2Fdspace.mit.edu%2Fbitstream%2Fhandle%2F1721.1%2F37335%2Ftech_report_ACL_TR-2007-001.pdf&ei=6THTUfakBKSviALF-IC4Cg&usg=AFQjCNFYjgxafbLe2wD79GB_xSrp0CciVA&sig2=NoRsXmiKxG7zeQIsdZjgIg&bvm=bv.48705608,d.cGE.

[15] William Green. *A Multimodal Micro Air Vehicle for Autonomous Flight in Near-Earth Environments*. PhD thesis, Drexel University, Philadelphia, PA,

June 2007. http://prism2.mem.drexel.edu/~billgreen/greenPhdThesis_052407.pdf.

[16] Eric Johnson, Michael Turbe, Allen Wu, Suresh Kannan, and James Neidhoefer. Flight test results of autonomous fixed-wing UAV transitions to and from stationary hover. AIAA Guidance, Navigation, and Control Conference Paper AIAA 2006-6775, Georgia Institute of Technology, Atlanta, Georgia, August 2006. https://smartech.gatech.edu/bitstream/handle/1853/35876/ejohnson_gnc_2006_28.pdf.

[17] Daisuke Kubo and Shinji Suzuki. Tail-sitter vertical takeoff and landing unmanned aerial vehicle: Transitional flight analysis. Journal of Aircraft Vol. 45, No. 1, University of Tokyo, January-February 2008. http://arc.aiaa.org/doi/abs/10.2514/1.30122?journalCode=ja.

[18] D. Kuchemann. A simple method for calculating the span and chordwise loading on straight and swept wings of any given aspect ratio at subsonic speeds. Aeronautical Research Council Reports and Memoranda 2935, Ministry of Supply, London: Her Majesty's Stationary Office, 1956. http://naca.central.cranfield.ac.uk/reports/arc/rm/2935.pdf.

[19] Christian Lopez. UAV formation flight utilizing a low cost, open source configuration. Master's thesis, California Polytechnic State University San Luis Obispo, San Luis Obispo, CA, June 2013.

[20] Takaaki Matsumoto, Koichi Kita, Ren Suzuki, Atsushi Oosedo, Kenta Go, Yuta Hoshino, Atsushi Konno, and Masaru Uchiyama. A hovering control strategy for a tail-sitter VTOL UAV that increases stability against large disturbance. IEEE International Conference on Robotics and Automation Paper 978-1-4244-5040-4, Tohoku University, Sendai, Miyagi Prefecture, Japan,

May 2010. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5509183&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5509183.

[21] Barnes McCormick, Jr. *Aerodynamics of V/STOL Flight*. Academic Press, New York, 1967.

[22] Eric Anthony Mehiel. *On Direct Model Reference Adaptive Controller Design For Flexible Space Structures*. PhD thesis, University of Colrado, Boulder, Boulder, CO, 2003.

[23] Joseph Moore and Russ Tedrake. Powerline perching with a fixed-wing UAV. Technical report, Massachusetts Institute of Technology, 2011. http://groups.csail.mit.edu/robotics-center/public_papers/Moore09.pdf.

[24] Vincent Myrand-Lapierre, Andre Desbiens, Eric Gagnon, Frank Wong, and Eric Poulin. Transitions between level flight and hovering for a fixed-wing mini aerial vehicle. IEEE American Control Conference Paper WeA15.3, Defence Research and Development Canada Valcartier, Qubec City, Qubec, July 2010. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5530875&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5530875.

[25] National Air and Space Museum, Smithsonian Institution.
Convair XFY-1 Pogo. Website, 2000. http://airandspace.si.edu/collections/artifact.cfm?id=A19730274000.

[26] Robert C. Nelson. *Flight Stability and Automatic Control*. The McGraw-Hill Companies, Inc., Boston, MA 1998.

[27] Net Resources International. Bell Eagle Eye Tiltrotor UAV, United States of America.

Website, 2012. http://www.naval-technology.com/projects/belleagleeyeuav/.

[28] Rushabh C. Patel. Adaptive output feedback control as applied to the rigid body equations of motion. Master's thesis, California Polytechnic State University San Luis Obispo, San Luis Obispo, CA, November 2007.

[29] Daniel Raymer. *Aircraft Design: A Conceptual Approach*. American Institute of Aeronautics and Astronautics, Washington, DC, second edition, 1992.

[30] Shane Alan Wallace. Development of a small and inexpensive terrain avoidance system for an unmanned aerial vehicle via potential function guidance algorithm. Master's thesis, California Polytechnic State University San Luis Obispo, San Luis Obispo, CA, September 2010.

[31] Derrick Yeo, Joshua Henderson, and Ella Atkins. An aerodynamic data system for small hovering fixed-wing UAS. AIAA Guidance, Navigation, and Control Conference Paper AIAA 2009-5756, University of Michigan, Ann Arbor, MI, August 2009. http://deepblue.lib.umich.edu/bitstream/handle/2027.42/76035/AIAA-2009-5756-435.pdf?sequence=1.

A SIMULATION RESULTS

Complete results for the 340 separate transition-to-hover simulations that were run as part of this thesis are presented here. Plots showing the time history of the aircraft's attitude, altitude, ground speed, and downrange distance traveled are included along with data for each simulation such as the wind speed, maximum altitude change, maximum distance traveled, and whether or not the airplane was able to achieve and maintain stable hover throughout the duration of the simulation.

Each of the three hover flight modes were tested under different conditions to see how well they handled different types of transitions. For the HOVER_PID_REF and HOVER_ADAPTIVE flight modes, each combination of controller type and reference model design was tested at four different approach speeds: 40, 60, 80, 100 ft/sec. Each controller, reference model, and approach speed combination was then repeated 5 times in order to get a measure of the repeatability of the results. Throttle controller input was always set to 0 ft/sec descent/climb rate for the entire duration of the simulations so variations in altitude are caused by either the bleeding off of excess kinetic energy or from the hover divergence logic controller needing to increase throttle in order to maintain stable hover, and not the pilot manually changing the desired descent/climb rate.

Each simulation was initiated with the aircraft pointing due North, with nose and wings level to the horizon (i.e. 0 degrees roll, pitch and yaw), at an altitude of 100 feet, and at the XY origin of the arbitrary Earth reference frame. Downrange distance traveled is the distance the aircraft travels only in the Earth's x-direction. Movement in the y-direction is ignored to eliminate the influence of lateral wind drift and produce comparable plots to the flight test results. The altitude change

is also plotted in reference to the starting altitude instead of the ground to make comparisons between simulation and flight test results easier. Additionally, ground speed is measured as the magnitude (independent of direction) of the velocity of the aircraft's center of mass in the XY plane (vertical velocity is ignored) to mimic the GPS based ground speed measurement taken during flight tests.

The colors that represent the plots for each of the 5 repeated tests per flight condition combination are as follows:

Flight No.	Plot Color	Wind Speed [ft/sec]
Ref. Model	Dashed Black	N/A
1	Solid Blue	0.0
2	Solid Red	5.0
3	Solid Green	10.0
4	Solid Magenta	15.0
5	Solid Cyan	20.0

Running the simulations all at the same wind speed would produce identical results among the 5 repeated tests, so a span of evenly spaced wind speeds between 0 and 20 ft/sec was used to test the controllers in both low and high speed wind conditions. 20 ft/sec was chosen as the upper limit because it is the approximate maximum wind speed where manually controlled RC flights are still safe and predictable.

Finally, a summary table is presented at the beginning of each controller type section which contains the probability of successful transition and the average, median, and standard deviation values for both the maximum altitude change, and maximum distance traveled for each combination of controller, reference model and approach speed.

A.1 PID Control Step Response

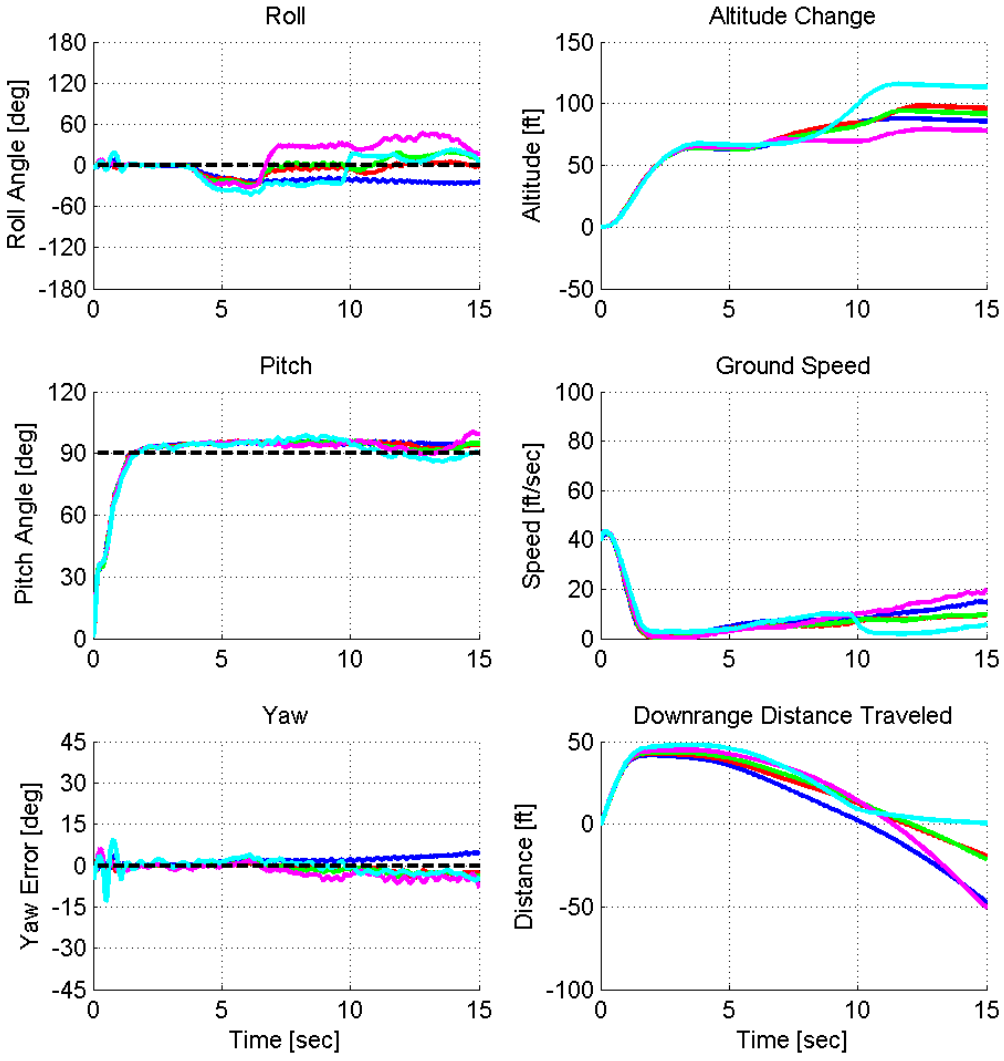
PID Control Step Response Simulations: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
Step	100%	100%	100%	100%	100%
Cumulative	100%	100%	100%	100%	100%

PID Control Step Response Simulations			
Approach Speed	Max. Altitude Change [ft]		
	Average	Median	Std. Dev.
40 ft/sec	95.2	94.3	13.6
60 ft/sec	111.0	110.9	11.2
80 ft/sec	152.1	158.3	11.0
100 ft/sec	180.2	181.0	5.4

PID Control Step Response Simulations			
Approach Speed	Max. Distance Traveled [ft]		
	Average	Median	Std. Dev.
40 ft/sec	44.1	43.5	2.4
60 ft/sec	48.4	48.5	0.6
80 ft/sec	58.6	58.8	1.8
100 ft/sec	64.2	64.2	2.4

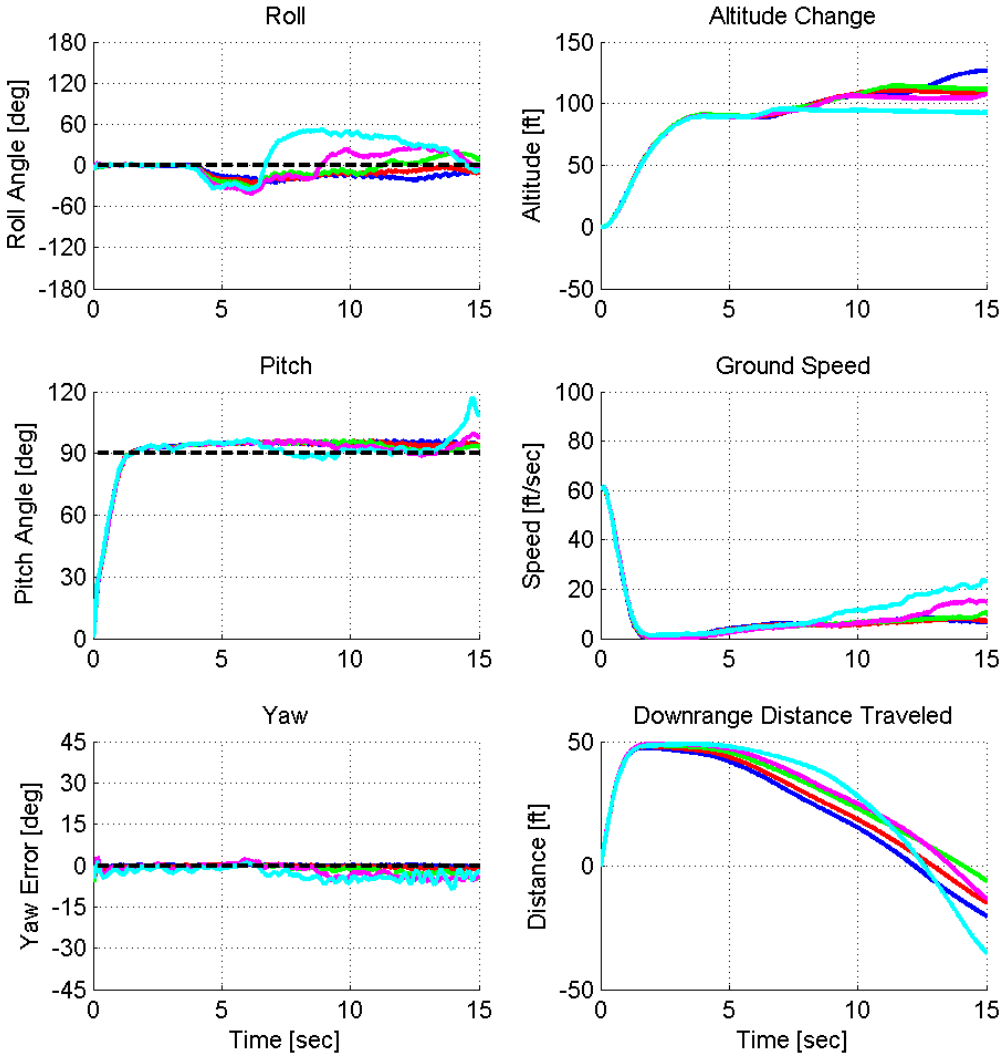
Simulation: PID Step Response

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	88.0	41.7
2	5.0	yes	98.3	42.7
3	10.0	yes	94.3	43.5
4	15.0	yes	79.3	44.9
5	20.0	yes	115.9	47.8
	Average		95.2	44.1
	Median		94.3	43.5
	Std. Dev.		13.6	2.4



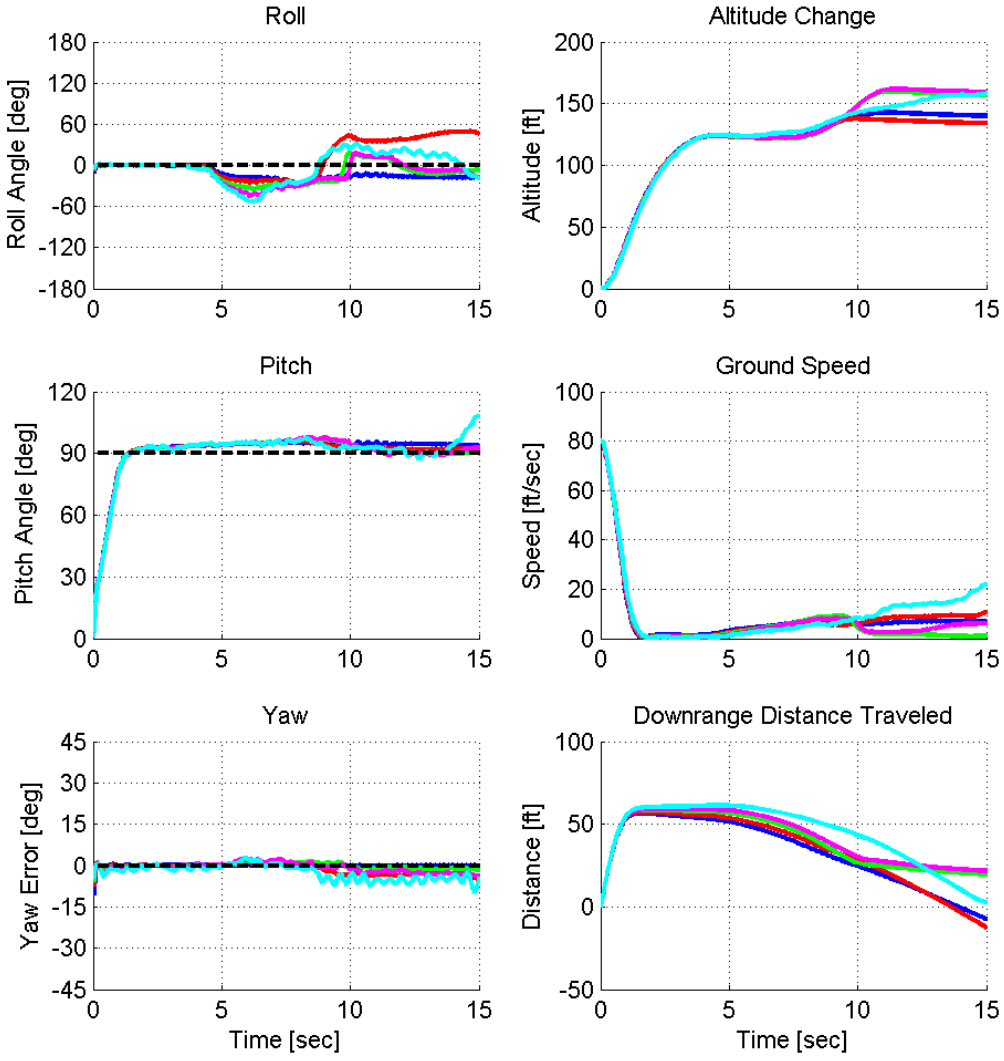
Simulation: PID Step Response

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	126.6	47.5
2	5.0	yes	110.9	48.0
3	10.0	yes	114.4	48.5
4	15.0	yes	107.3	49.0
5	20.0	yes	95.7	48.9
	Average		110.0	48.4
	Median		110.9	48.5
	Std. Dev.		11.2	0.6



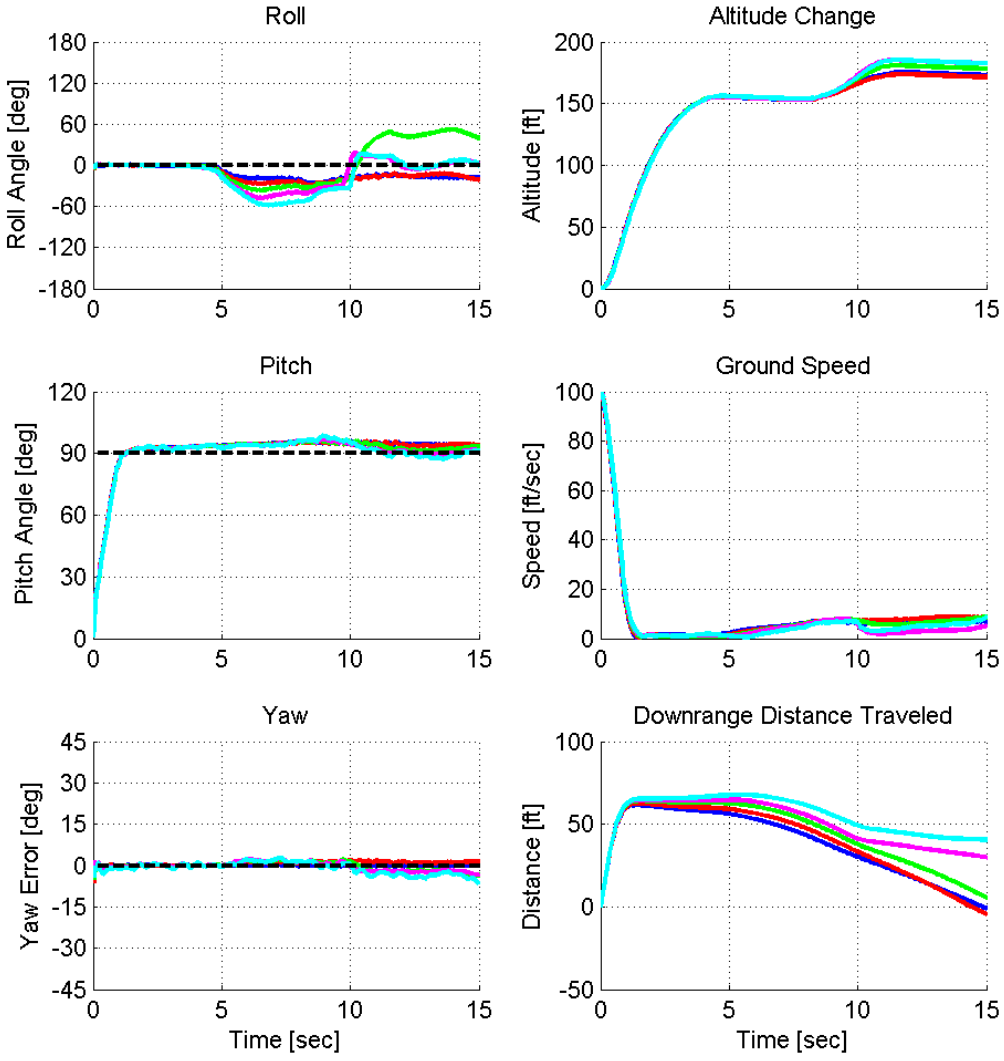
Simulation: PID Step Response

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	142.8	56.6
2	5.0	yes	137.8	57.4
3	10.0	yes	159.9	58.8
4	15.0	yes	161.9	59.0
5	20.0	yes	158.3	61.3
	Average		152.1	58.6
	Median		158.3	58.8
	Std. Dev.		11.0	1.8



Simulation: PID Step Response

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	175.3	61.5
2	5.0	yes	173.8	62.6
3	10.0	yes	181.0	64.2
4	15.0	yes	185.4	64.8
5	20.0	yes	185.4	67.7
	Average		180.2	64.2
	Median		181.0	64.2
	Std. Dev.		5.4	2.4



A.2 PID Control Reference Model Response

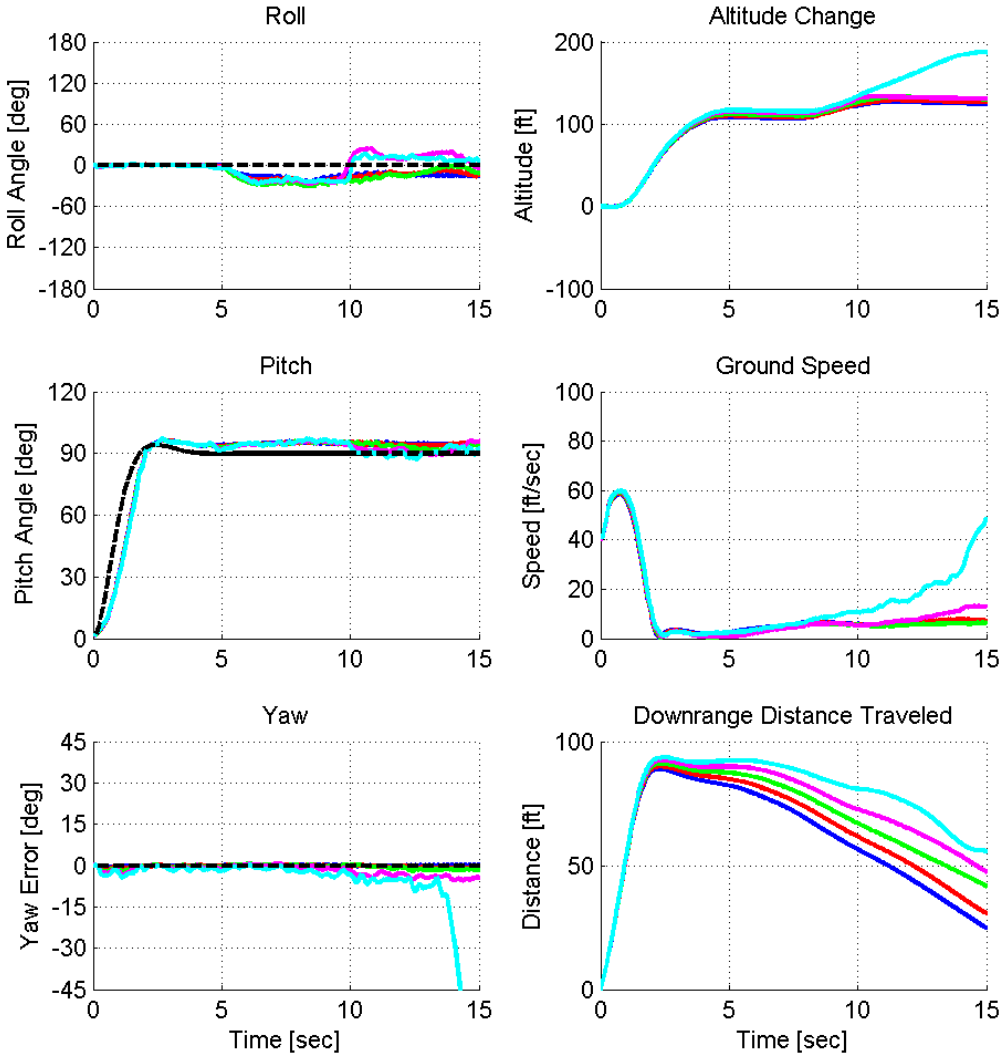
PID Control Reference Model Response Simulations: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
5	80%	60%	80%	40%	65%
7	40%	20%	20%	20%	35%
10	20%	0%	40%	0%	15%
15	40%	0%	0%	20%	15%
20	0%	0%	0%	0%	0%
Cumulative	60%	48%	55%	47%	53%

PID Control Reference Model Response Simulations				
Rise Time	Approach Speed	Max. Altitude Change [ft]		
		Average	Median	Std. Dev.
1 sec	40 ft/sec	142.3	133.3	25.4
	60 ft/sec	142.4	141.7	4.9
	80 ft/sec	156.0	156.4	1.6
	100 ft/sec	174.0	174.6	2.3
2 sec	40 ft/sec	169.0	169.1	10.4
	60 ft/sec	173.9	177.0	10.5
	80 ft/sec	184.7	188.8	9.1
	100 ft/sec	206.2	208.6	8.3
3 sec	40 ft/sec	191.1	194.2	6.3
	60 ft/sec	192.7	195.3	6.6
	80 ft/sec	201.8	205.2	6.9
	100 ft/sec	217.2	218.0	2.9
5 sec	40 ft/sec	236.7	237.5	29.5
	60 ft/sec	231.2	249.3	40.3
	80 ft/sec	229.9	237.6	32.9
	100 ft/sec	268.0	268.0	6.6
7 sec	40 ft/sec	589.0	589.0	417.7
	60 ft/sec	315.2	315.2	N/A
	80 ft/sec	326.8	326.8	N/A
	100 ft/sec	323.4	323.4	N/A
10 sec	40 ft/sec	740.2	740.2	N/A
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	571.6	571.6	300.0
	100 ft/sec	N/A	N/A	N/A
15 sec	40 ft/sec	727.9	727.9	386.0
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	851.7	851.7	N/A
20 sec	40 ft/sec	N/A	N/A	N/A
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	N/A	N/A	N/A

PID Control Reference Model Response Simulations				
Rise Time	Approach Speed	Max. Distance Traveled [ft]		
		Average	Median	Std. Dev.
1 sec	40 ft/sec	91.2	91.1	1.9
	60 ft/sec	104.2	104.0	3.0
	80 ft/sec	120.5	120.1	1.9
	100 ft/sec	136.3	136.2	1.4
2 sec	40 ft/sec	162.9	162.8	1.9
	60 ft/sec	176.1	174.9	2.9
	80 ft/sec	194.6	193.7	3.4
	100 ft/sec	215.2	214.8	1.8
3 sec	40 ft/sec	239.0	238.7	3.7
	60 ft/sec	248.5	250.2	3.5
	80 ft/sec	263.8	263.0	3.6
	100 ft/sec	286.6	286.3	1.3
5 sec	40 ft/sec	384.9	386.5	7.7
	60 ft/sec	395.7	394.6	2.2
	80 ft/sec	404.7	405.8	6.1
	100 ft/sec	432.4	432.4	4.0
7 sec	40 ft/sec	486.3	486.3	12.8
	60 ft/sec	499.1	499.1	N/A
	80 ft/sec	512.2	512.2	N/A
	100 ft/sec	536.8	536.8	N/A
10 sec	40 ft/sec	669.1	669.1	N/A
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	709.5	709.5	16.7
	100 ft/sec	N/A	N/A	N/A
15 sec	40 ft/sec	924.8	924.8	23.2
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	964.4	964.4	N/A
20 sec	40 ft/sec	N/A	N/A	N/A
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	N/A	N/A	N/A

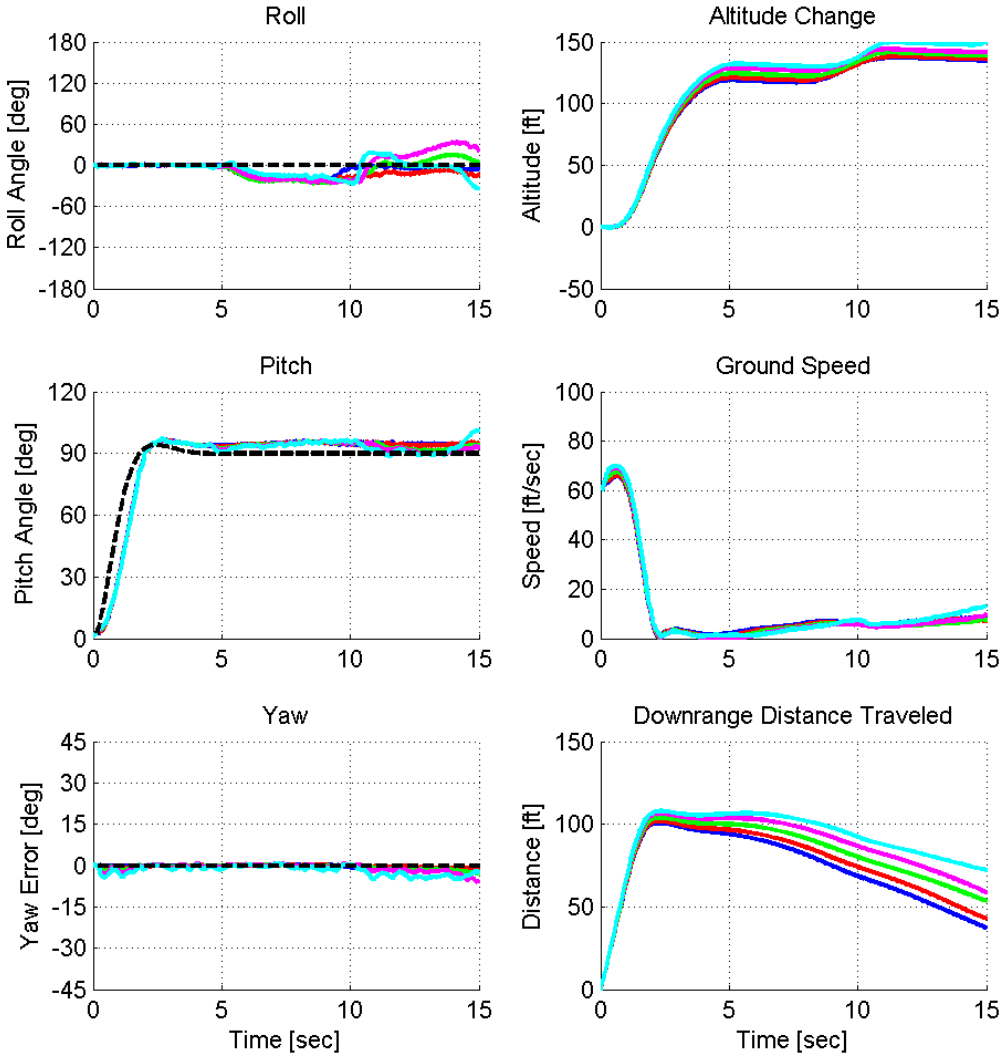
Simulation: PID Reference Model Response, Rise Time = 1 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	127.4	88.9
2	5.0	yes	129.5	90.0
3	10.0	yes	133.3	91.1
4	15.0	yes	133.8	92.4
5	20.0	yes	187.5	93.6
	Average		142.3	91.2
	Median		133.3	91.1
	Std. Dev.		25.4	1.9



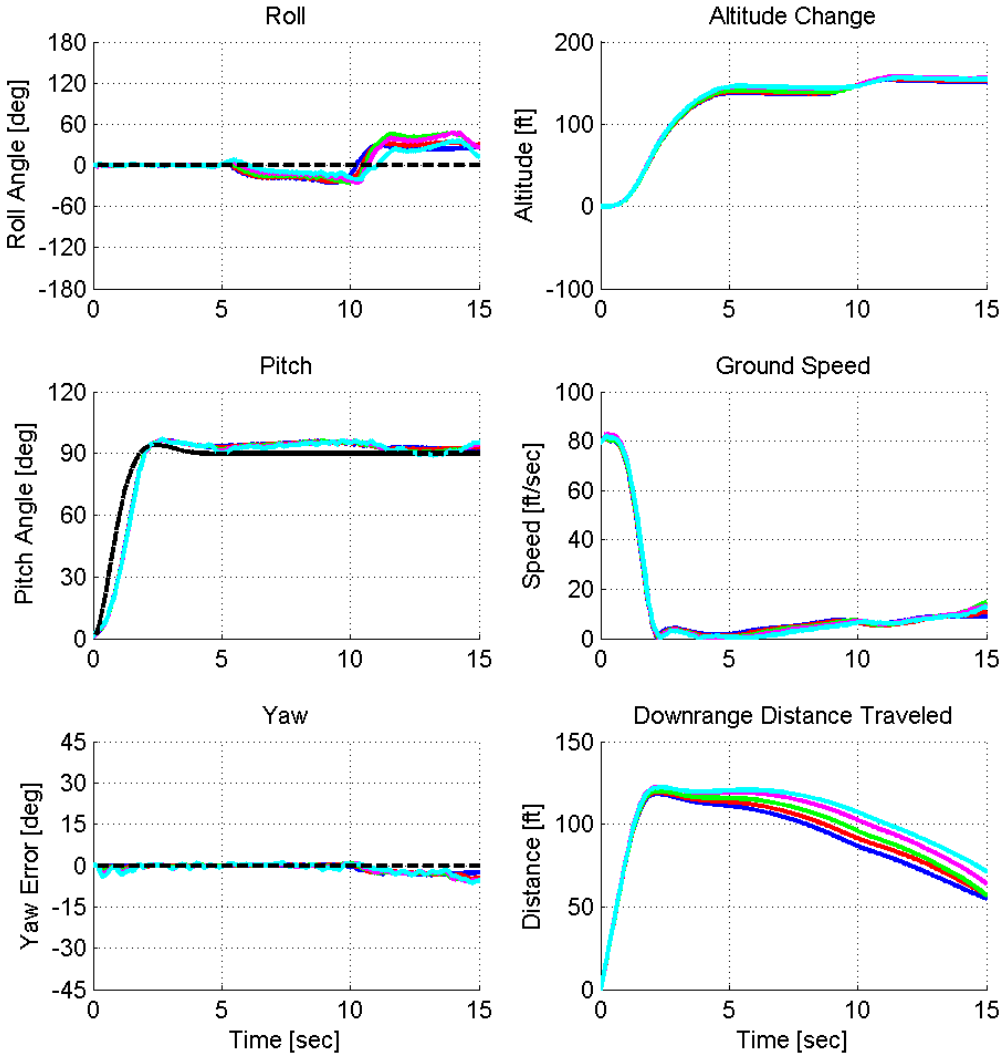
Simulation: PID Reference Model Response, Rise Time = 1 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	137.7	100.8
2	5.0	yes	138.6	101.9
3	10.0	yes	141.7	104.0
4	15.0	yes	144.2	106.4
5	20.0	yes	149.9	107.8
	Average		142.4	104.2
	Median		141.7	104.0
	Std. Dev.		4.9	3.0



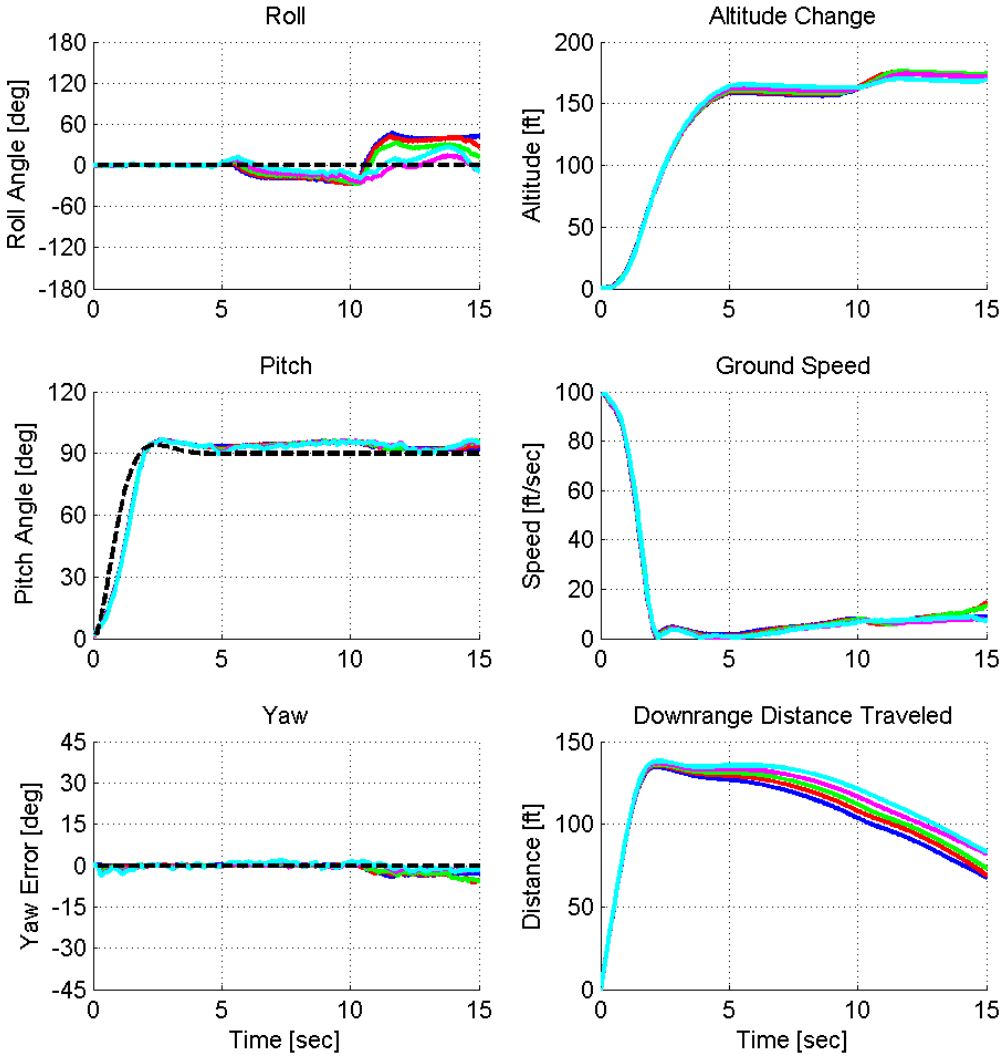
Simulation: PID Reference Model Response, Rise Time = 1 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	153.9	118.3
2	5.0	yes	155.1	119.2
3	10.0	yes	157.3	120.1
4	15.0	yes	157.6	122.5
5	20.0	yes	156.4	122.3
	Average		156.0	120.5
	Median		156.4	120.1
	Std. Dev.		1.6	1.9



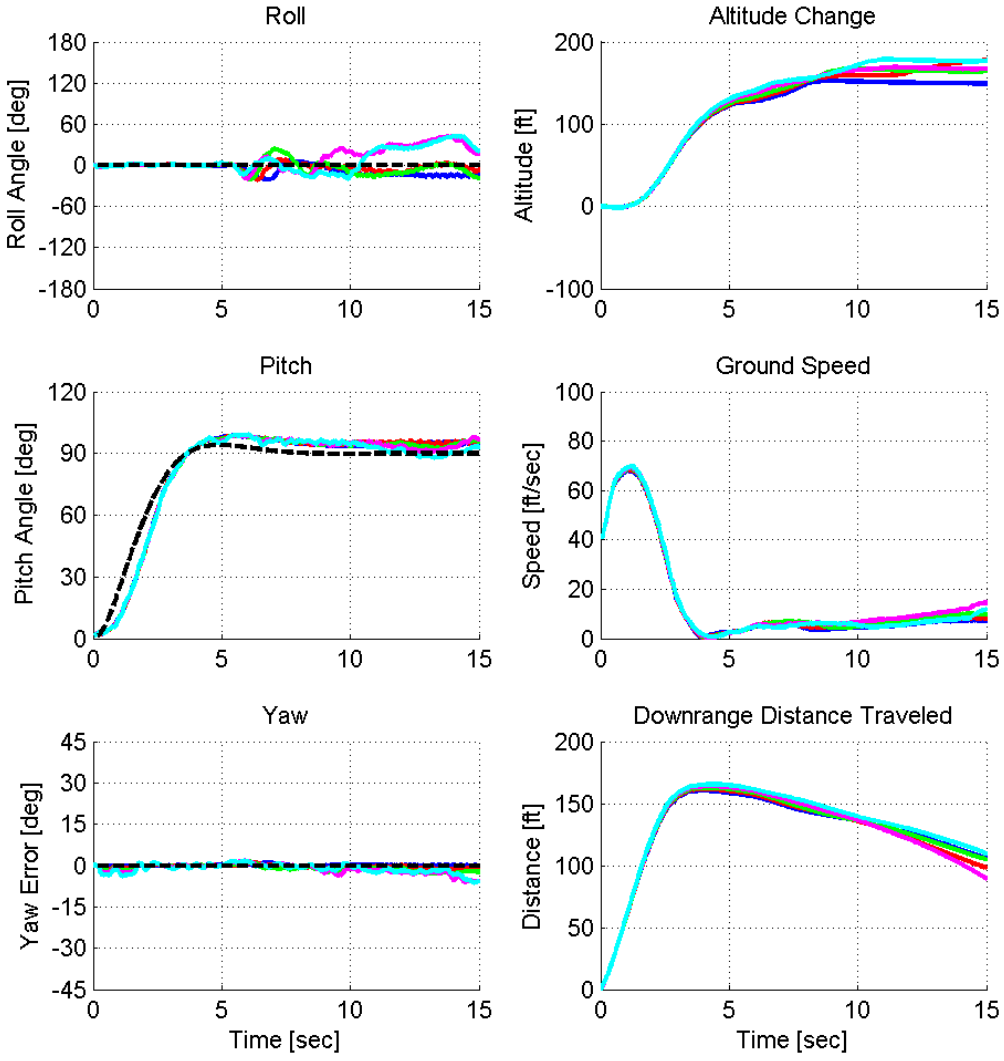
Simulation: PID Reference Model Response, Rise Time = 1 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	174.6	134.5
2	5.0	yes	175.2	135.5
3	10.0	yes	176.0	136.2
4	15.0	yes	174.1	136.9
5	20.0	yes	170.1	138.1
	Average		174.0	136.3
	Median		174.6	136.2
	Std. Dev.		2.3	1.4



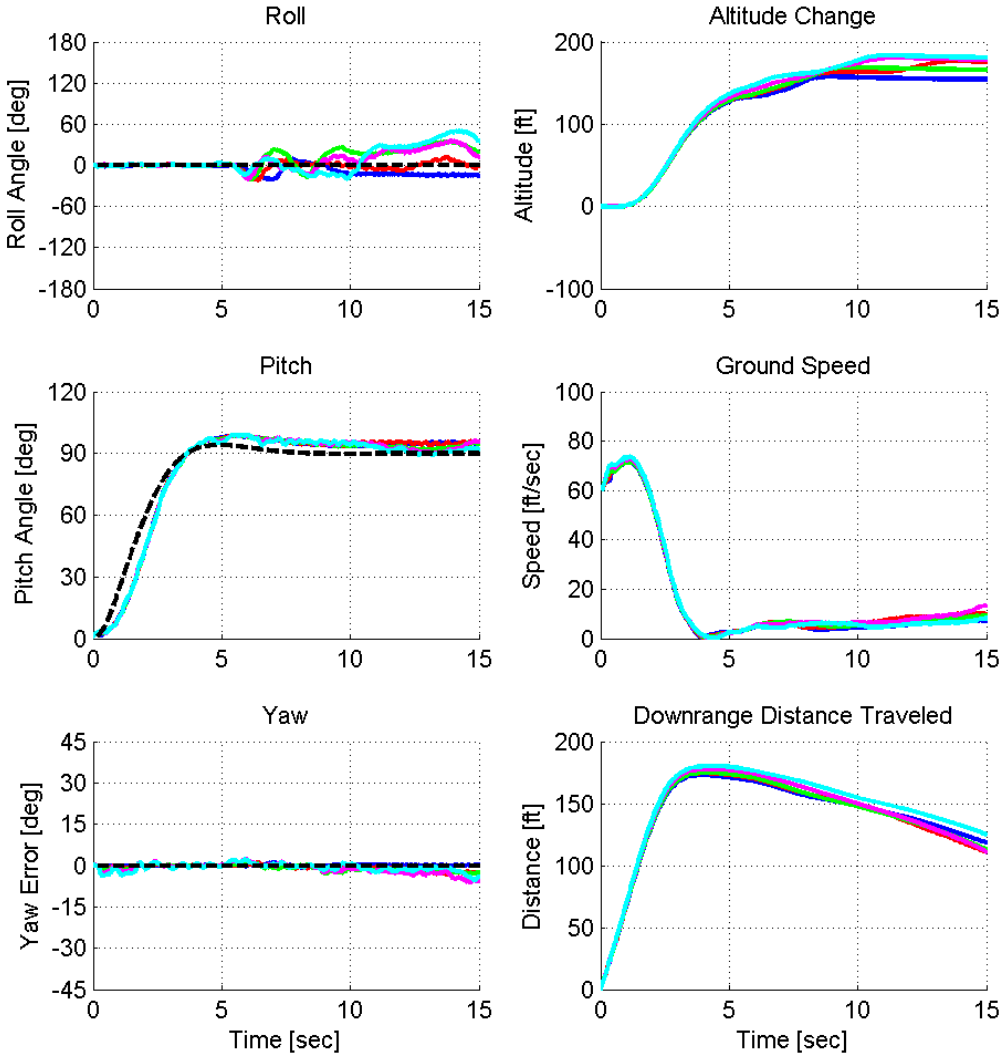
Simulation: PID Reference Model Response, Rise Time = 2 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	152.8	160.4
2	5.0	yes	177.6	161.7
3	10.0	yes	166.9	162.8
4	15.0	yes	169.1	164.0
5	20.0	yes	178.7	165.3
	Average		169.0	162.9
	Median		169.1	162.8
	Std. Dev.		10.4	1.9



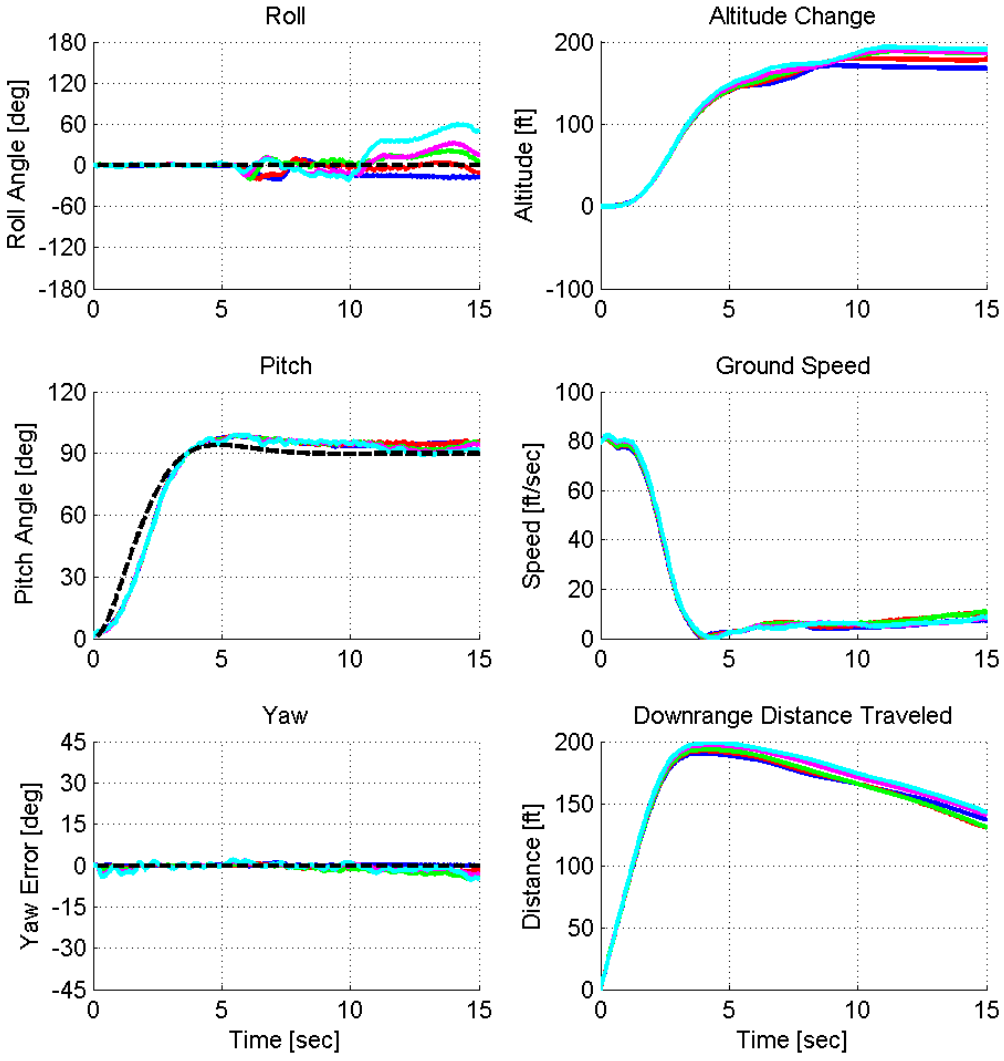
Simulation: PID Reference Model Response, Rise Time = 2 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	157.8	173.0
2	5.0	yes	177.0	174.6
3	10.0	yes	169.5	174.9
4	15.0	yes	181.2	177.9
5	20.0	yes	183.8	180.2
	Average		173.9	176.1
	Median		177.0	174.9
	Std. Dev.		10.5	2.9



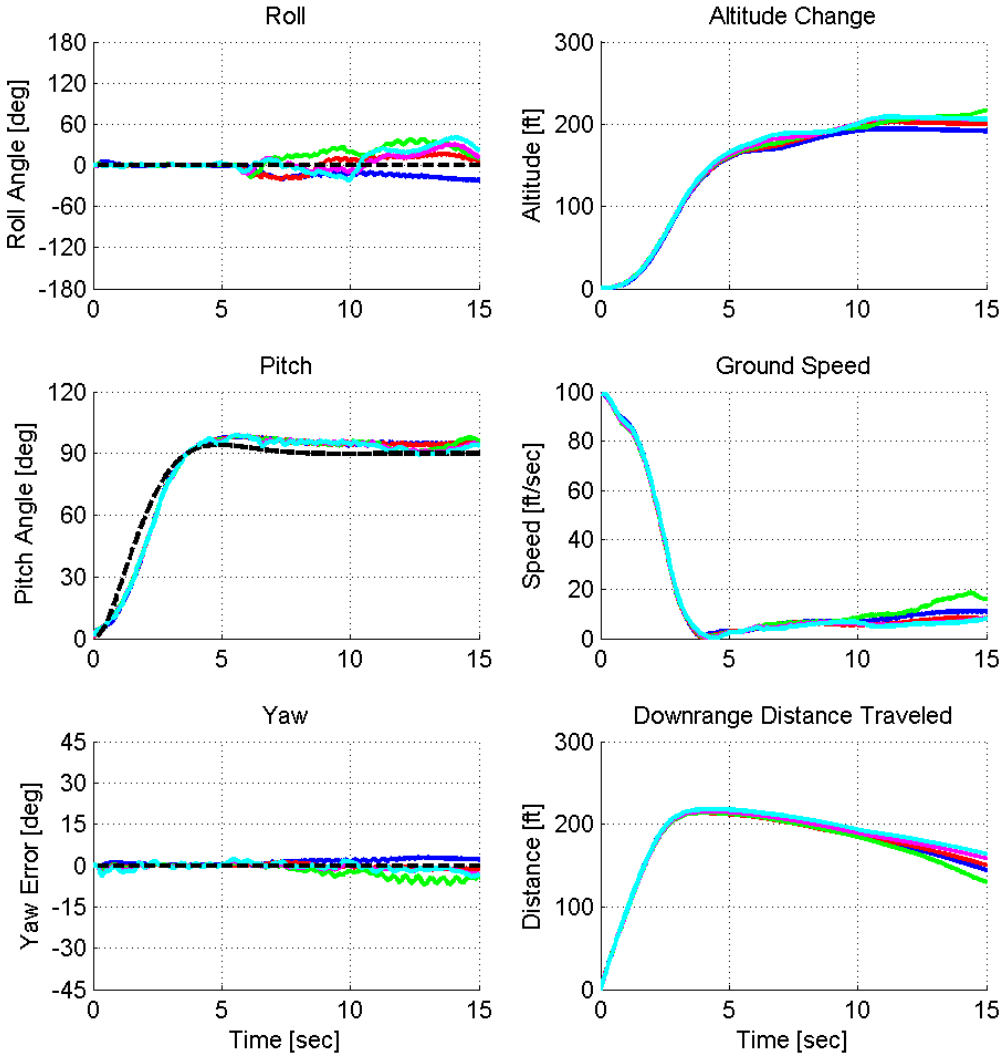
Simulation: PID Reference Model Response, Rise Time = 2 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	171.3	190.6
2	5.0	yes	179.8	192.5
3	10.0	yes	188.8	193.7
4	15.0	yes	190.0	197.1
5	20.0	yes	193.6	198.9
	Average		184.7	194.6
	Median		188.8	193.7
	Std. Dev.		9.1	3.4



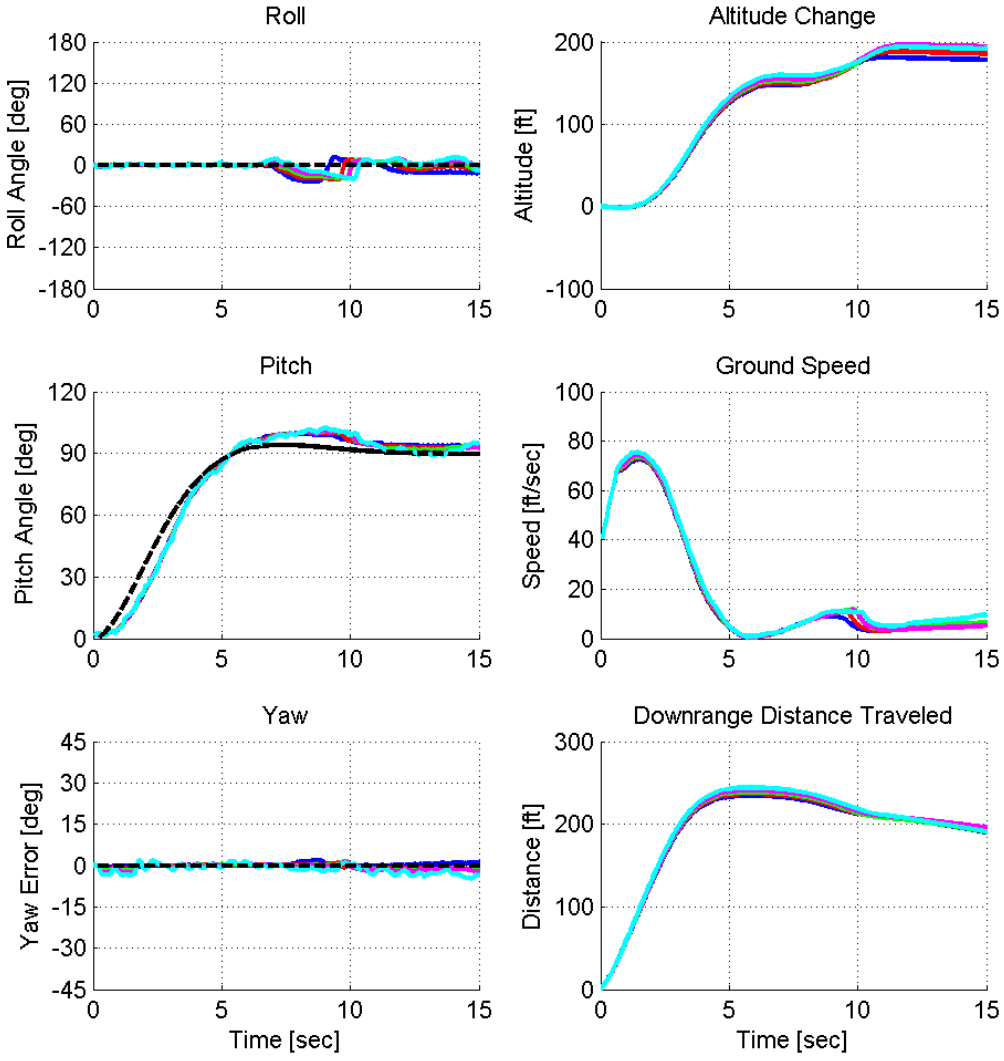
Simulation: PID Reference Model Response, Rise Time = 2 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	194.3	214.8
2	5.0	yes	202.7	213.7
3	10.0	yes	216.6	213.9
4	15.0	yes	208.6	215.6
5	20.0	yes	208.9	218.2
	Average		206.2	215.2
	Median		208.6	214.8
	Std. Dev.		8.3	1.8



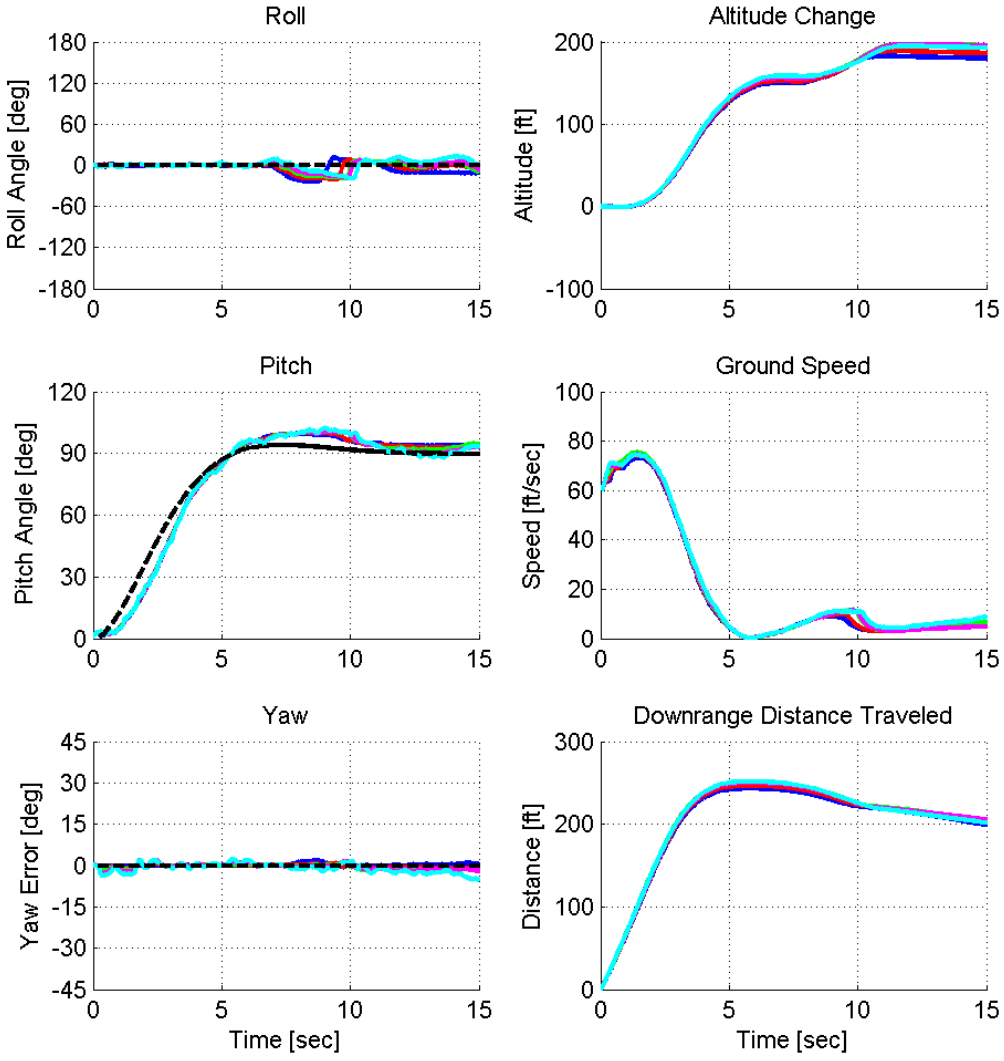
Simulation: PID Reference Model Response, Rise Time = 3 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	181.2	234.5
2	5.0	yes	188.4	236.7
3	10.0	yes	194.7	238.7
4	15.0	yes	196.9	240.9
5	20.0	yes	194.2	244.1
	Average		191.1	239.0
	Median		194.2	238.7
	Std. Dev.		6.3	3.7



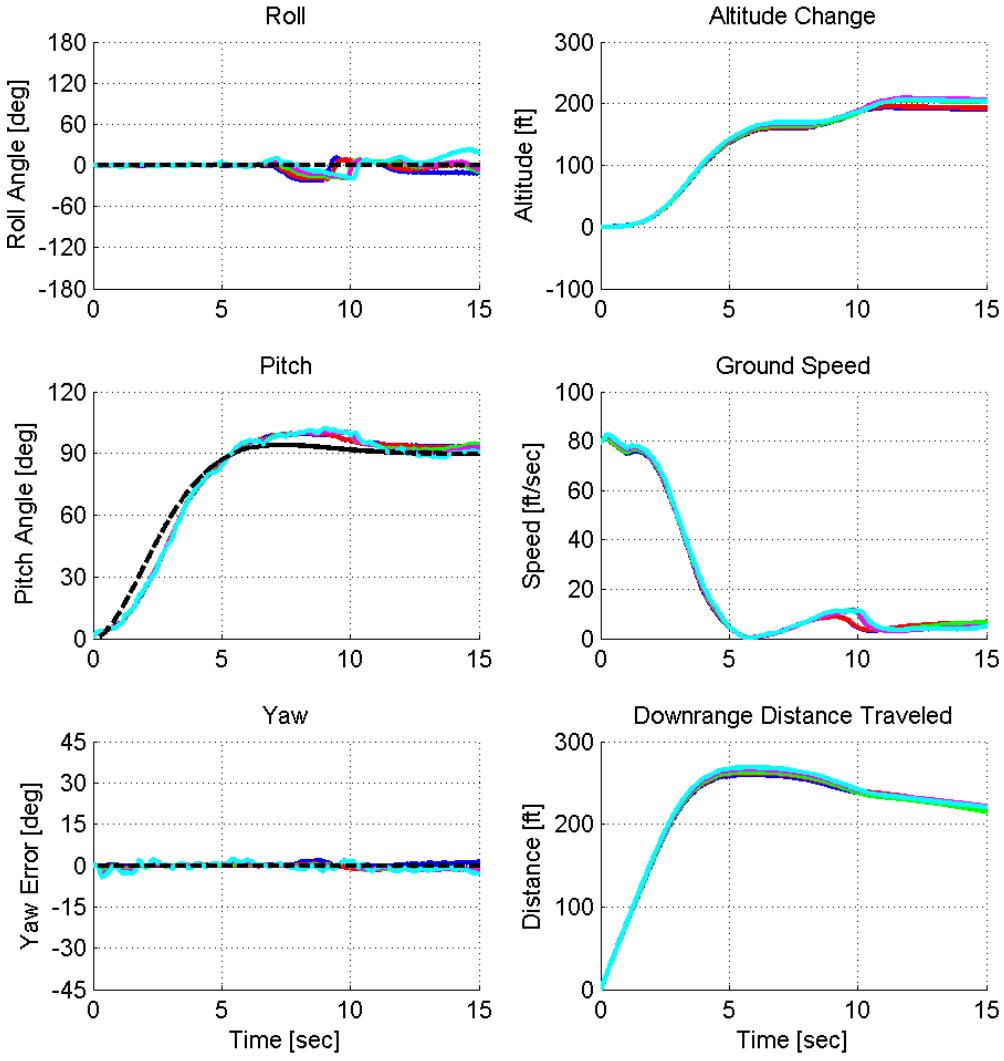
Simulation: PID Reference Model Response, Rise Time = 3 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	182.9	243.3
2	5.0	yes	189.1	246.6
3	10.0	yes	198.2	250.8
4	15.0	yes	197.8	250.2
5	20.0	yes	195.3	251.5
	Average		192.7	248.5
	Median		195.3	250.2
	Std. Dev.		6.6	3.5



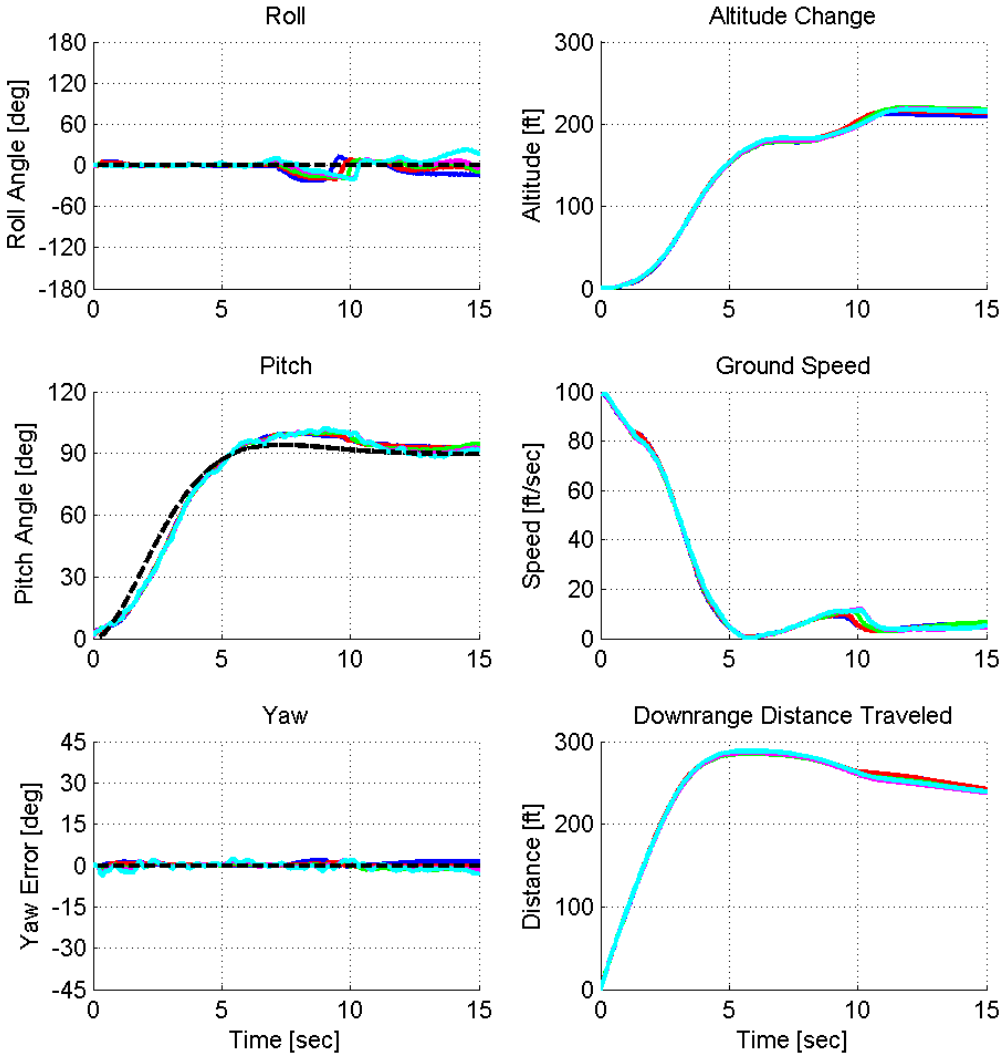
Simulation: PID Reference Model Response, Rise Time = 3 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	193.5	259.6
2	5.0	yes	195.5	261.8
3	10.0	yes	205.2	263.0
4	15.0	yes	208.8	266.2
5	20.0	yes	206.3	268.6
	Average		201.8	263.8
	Median		205.2	263.0
	Std. Dev.		6.9	3.6



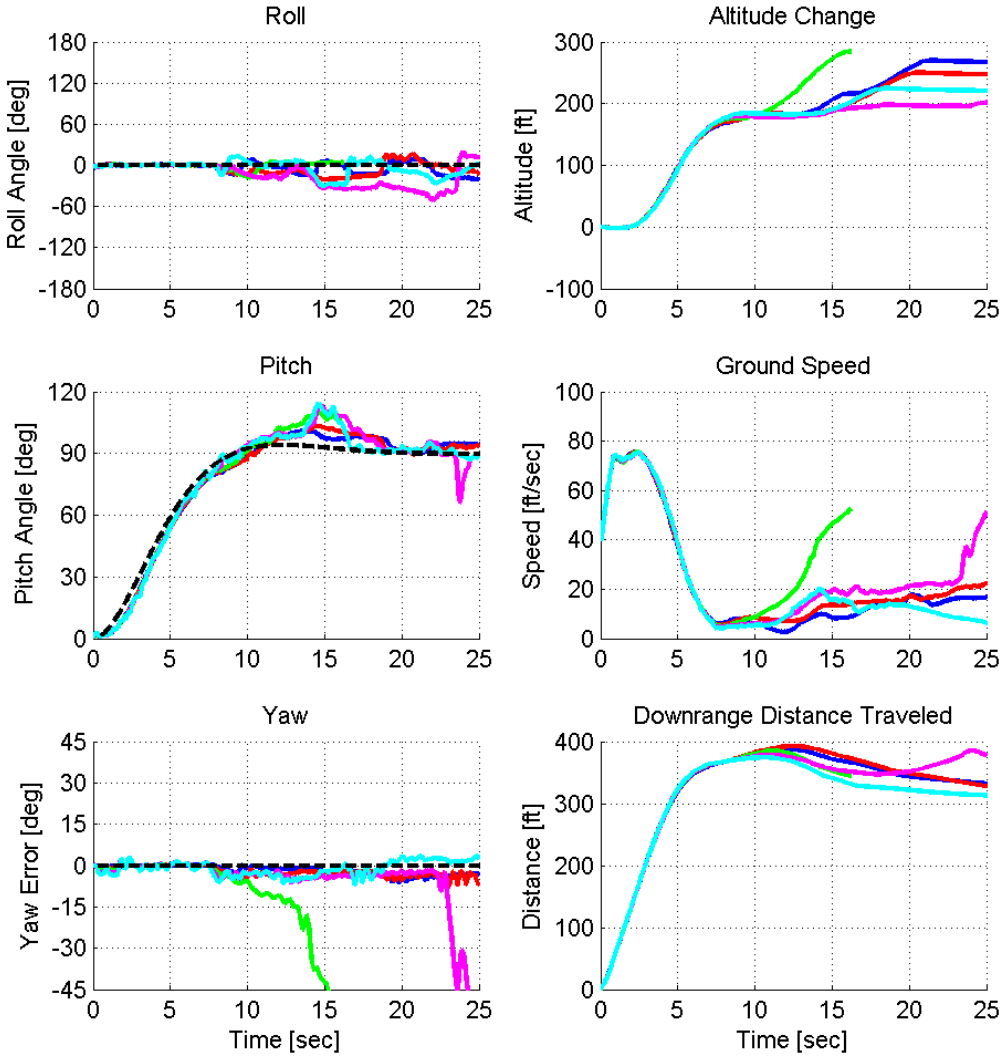
Simulation: PID Reference Model Response, Rise Time = 3 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	212.7	285.9
2	5.0	yes	216.5	287.2
3	10.0	yes	220.3	285.2
4	15.0	yes	218.7	286.3
5	20.0	yes	218.0	288.5
	Average		217.2	286.6
	Median		218.0	286.3
	Std. Dev.		2.9	1.3



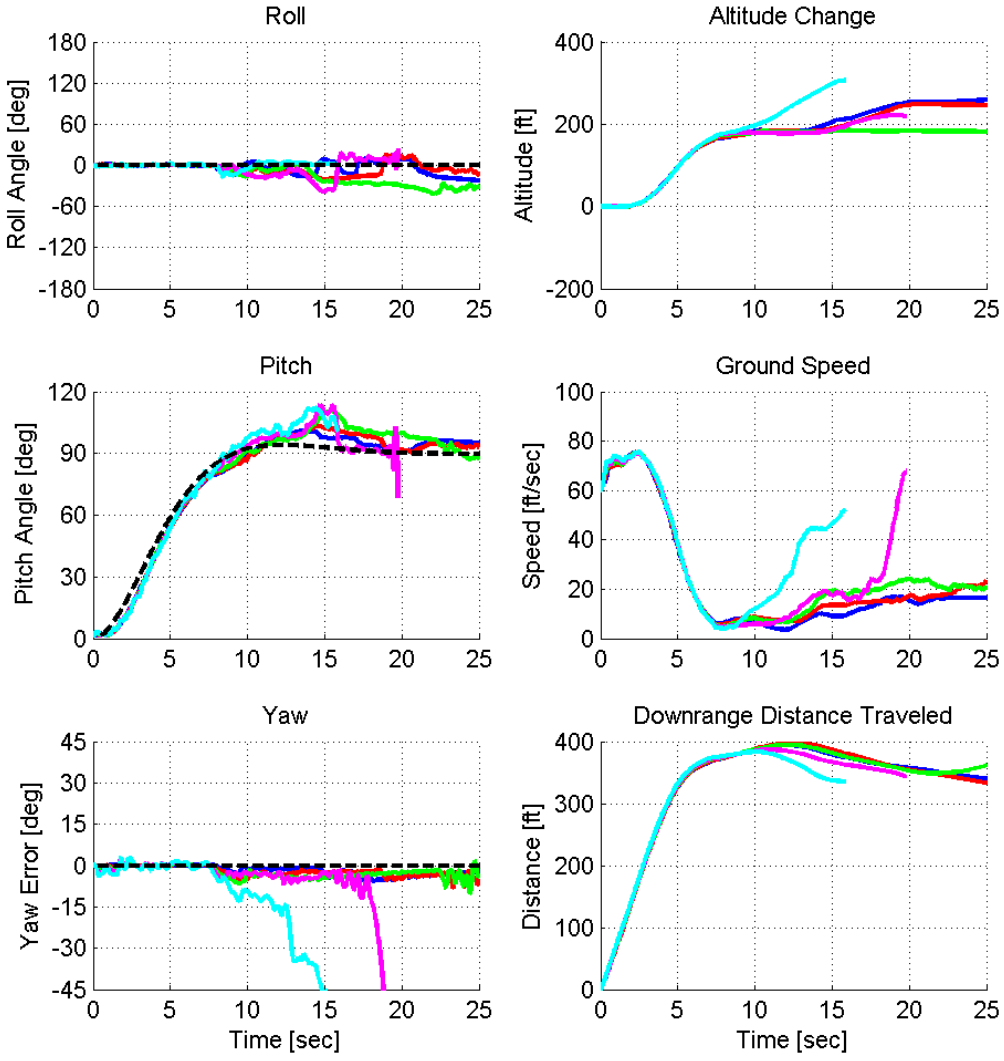
Simulation: PID Reference Model Response, Rise Time = 5 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	269.7	387.7
2	5.0	yes	250.2	392.4
3	10.0	no	284.7	384.8
4	15.0	yes	202.1	385.3
5	20.0	yes	224.9	374.2
	Average		236.7	384.9
	Median		237.5	386.5
	Std. Dev.		29.5	7.7



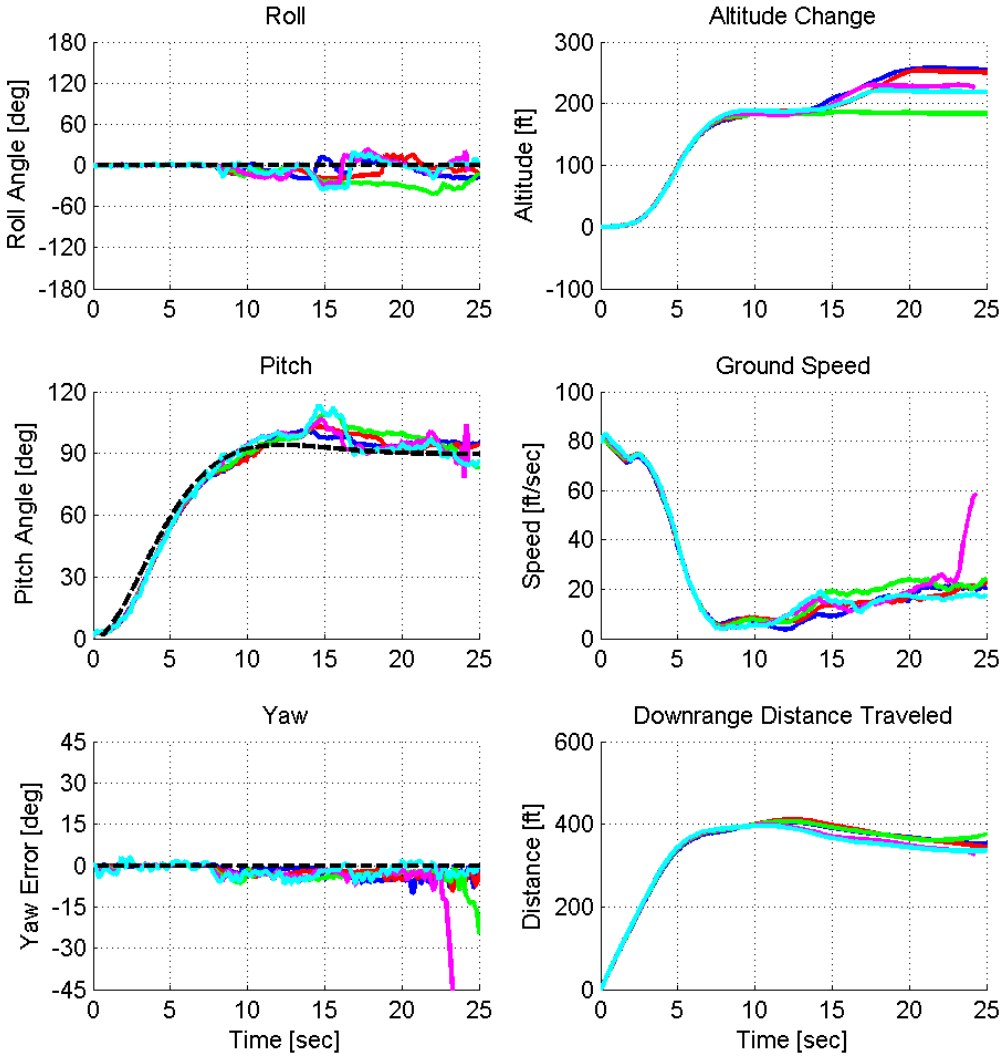
Simulation: PID Reference Model Response, Rise Time = 5 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	259.2	394.2
2	5.0	yes	249.3	398.2
3	10.0	yes	185.0	394.6
4	15.0	no	222.0	386.7
5	20.0	no	305.9	383.3
	Average		231.2	395.7
	Median		249.3	394.6
	Std. Dev.		40.3	2.2



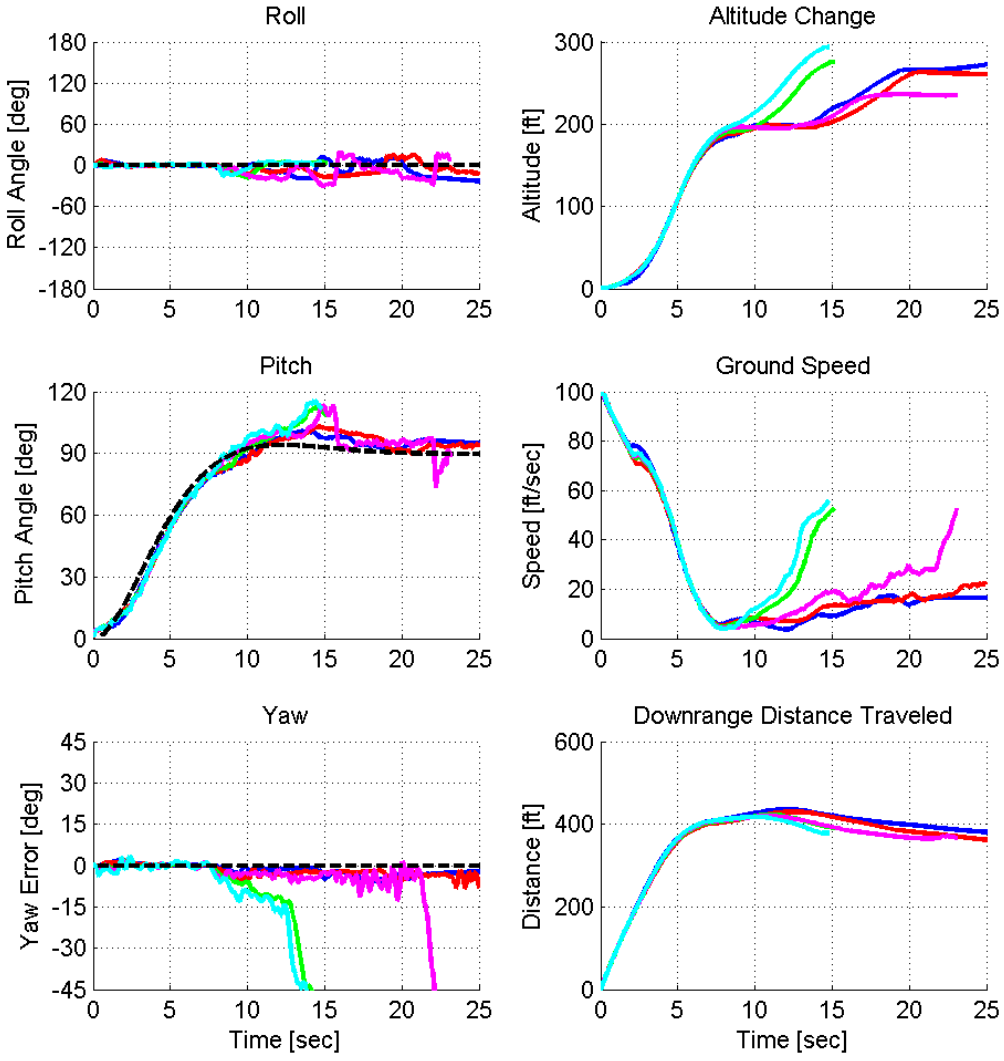
Simulation: PID Reference Model Response, Rise Time = 5 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	257.6	405.3
2	5.0	yes	253.6	410.9
3	10.0	yes	186.9	406.3
4	15.0	no	230.2	397.4
5	20.0	yes	221.7	396.2
	Average		229.9	404.7
	Median		237.6	405.8
	Std. Dev.		32.9	6.1



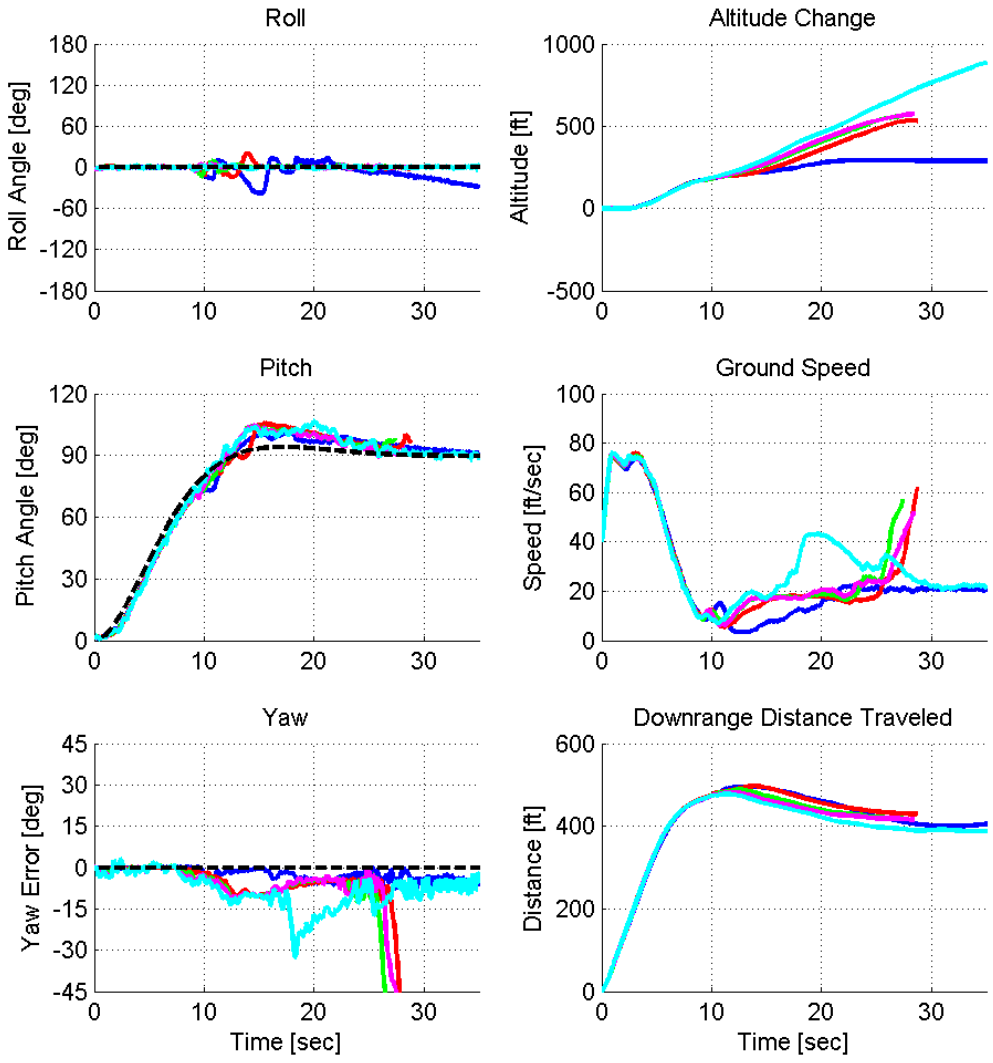
Simulation: PID Reference Model Response, Rise Time = 5 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	272.6	435.2
2	5.0	yes	263.3	429.6
3	10.0	no	276.0	422.7
4	15.0	no	236.4	418.7
5	20.0	no	293.8	417.8
	Average		268.0	432.4
	Median		268.0	432.4
	Std. Dev.		6.6	4.0



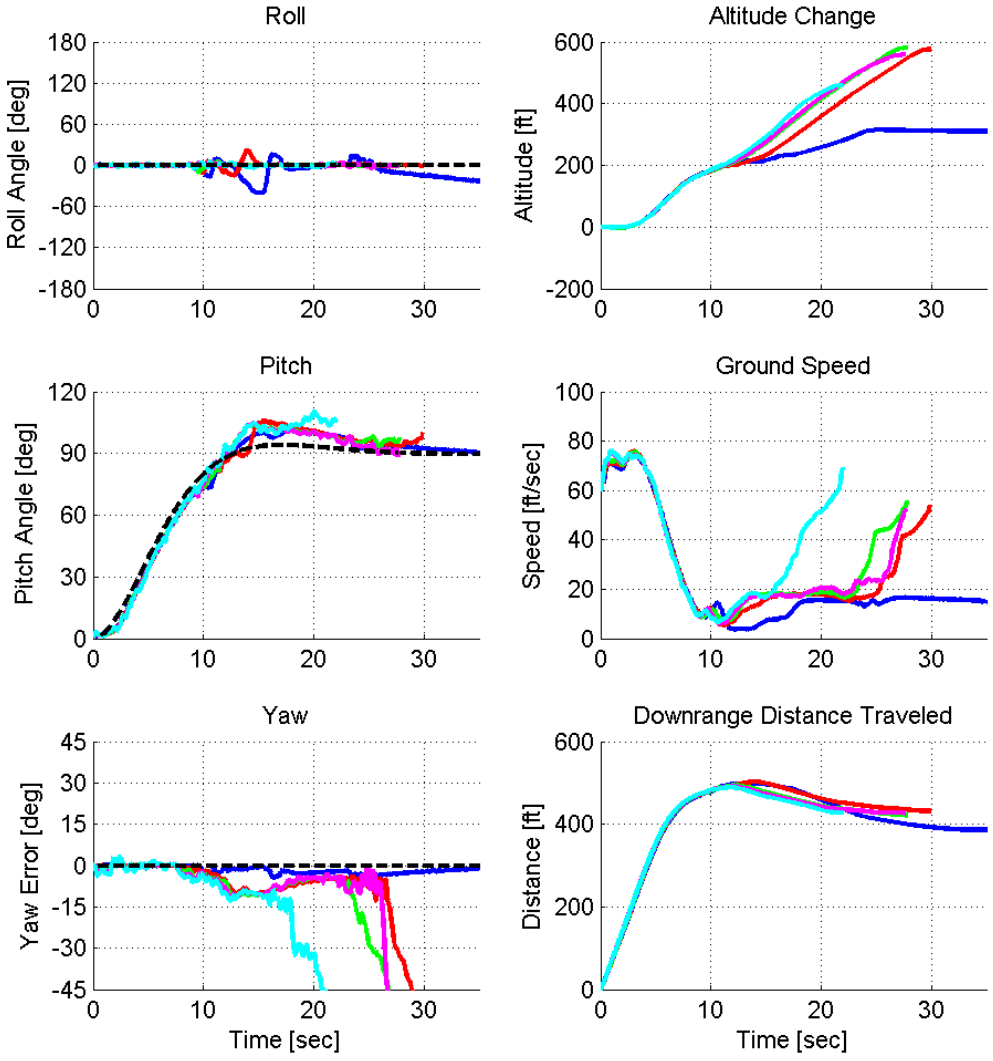
Simulation: PID Reference Model Response, Rise Time = 7 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	293.7	495.4
2	5.0	no	534.1	497.0
3	10.0	no	559.8	488.2
4	15.0	no	573.0	481.4
5	20.0	yes	884.3	477.2
	Average		589.0	486.3
	Median		589.0	486.3
	Std. Dev.		417.7	12.8



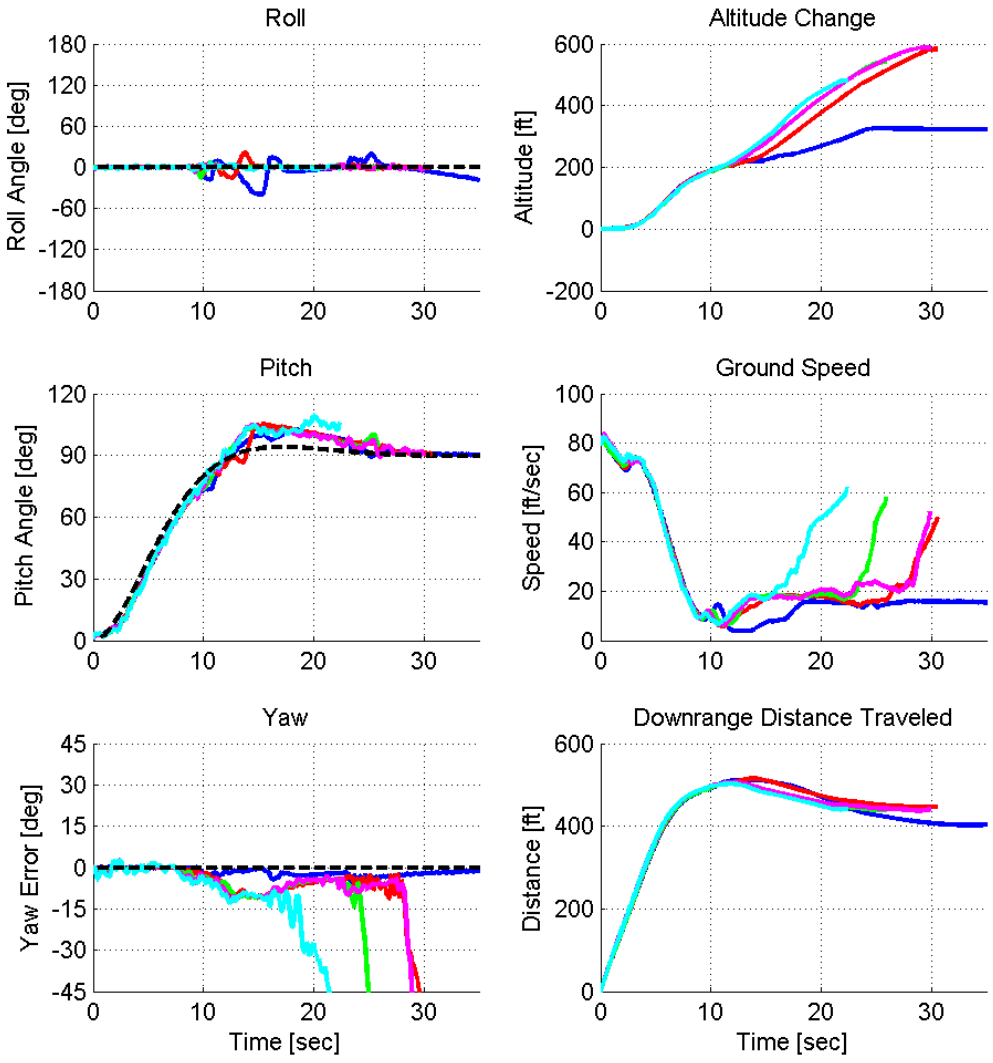
Simulation: PID Reference Model Response, Rise Time = 7 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	315.2	499.1
2	5.0	no	575.8	502.7
3	10.0	no	580.9	493.4
4	15.0	no	559.1	490.7
5	20.0	no	458.9	489.0
	Average		315.2	499.1
	Median		315.2	499.1
	Std. Dev.		N/A	N/A



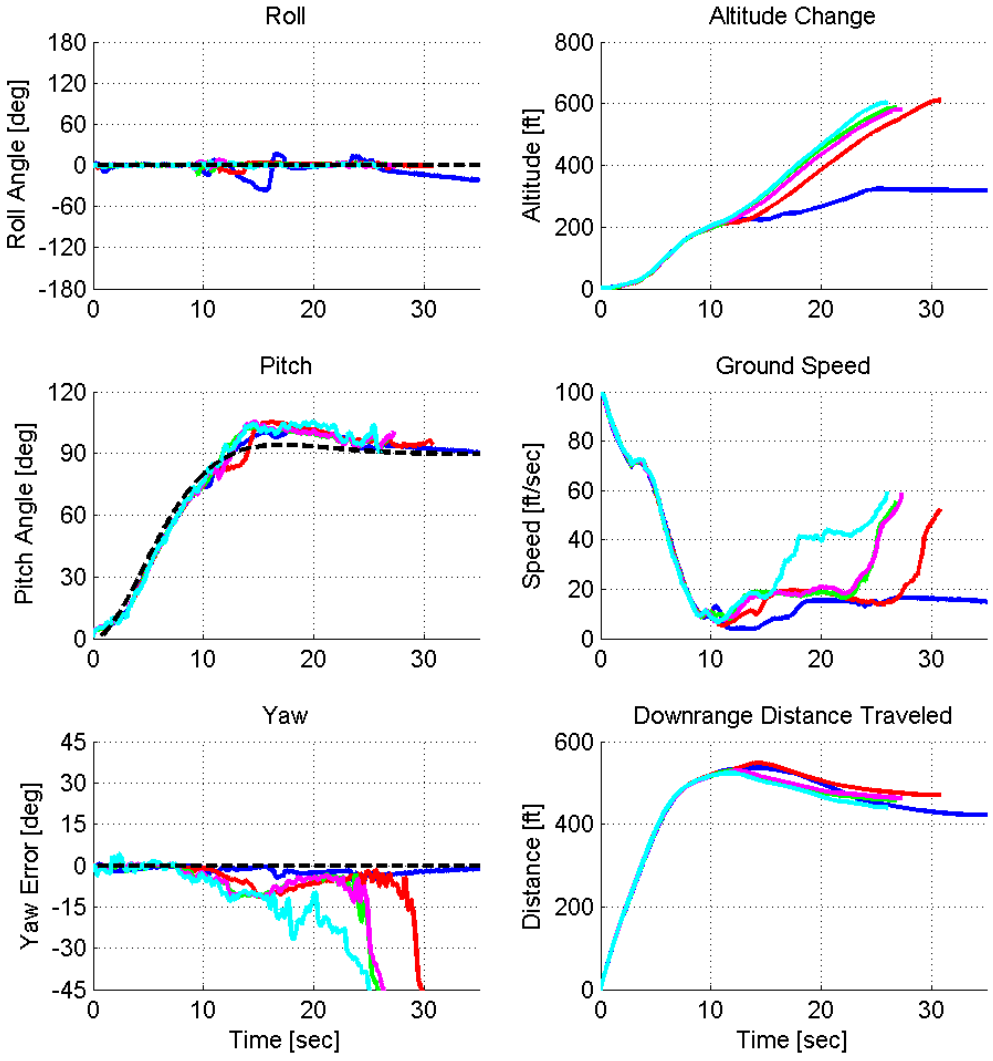
Simulation: PID Reference Model Response, Rise Time = 7 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	326.8	512.2
2	5.0	no	583.9	512.2
3	10.0	no	541.4	504.5
4	15.0	no	589.3	505.8
5	20.0	no	480.8	502.5
	Average		326.8	512.2
	Median		326.8	512.2
	Std. Dev.		N/A	N/A



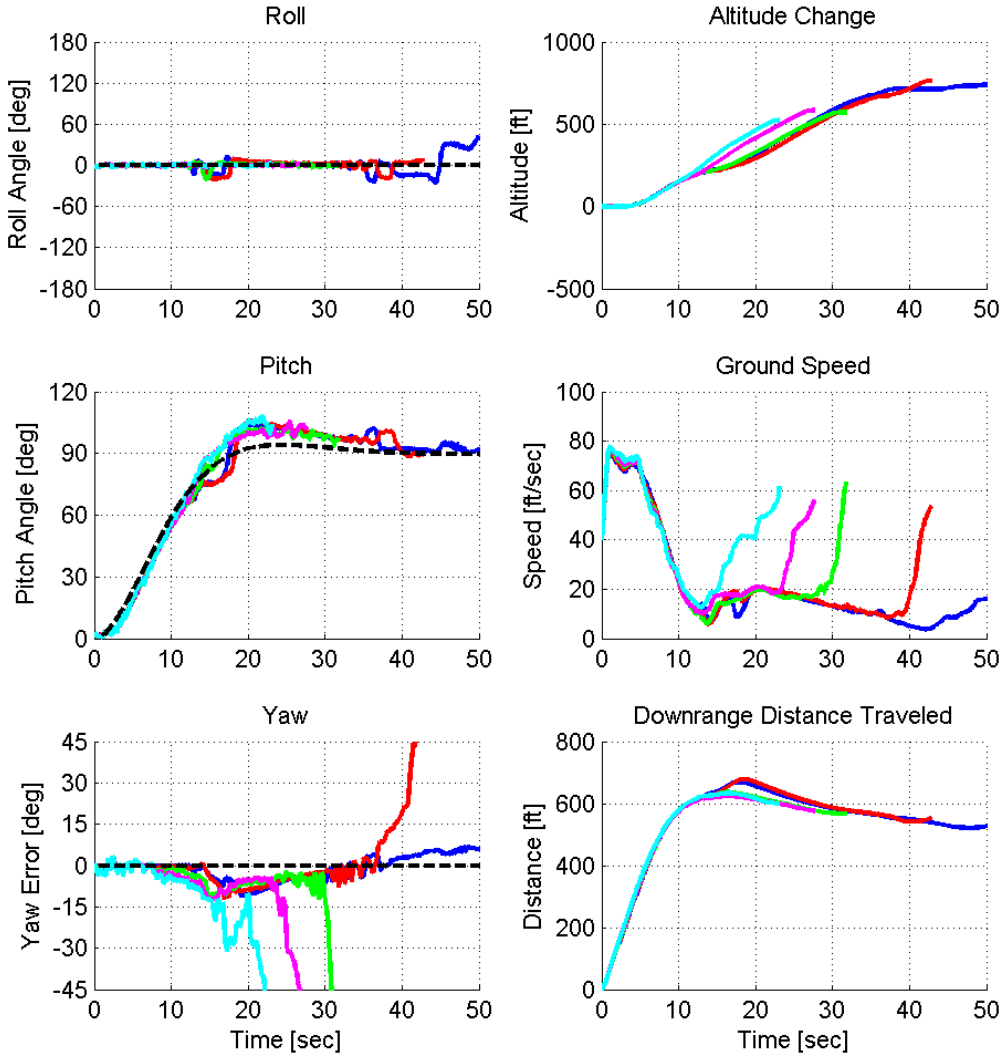
Simulation: PID Reference Model Response, Rise Time = 7 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	323.4	536.8
2	5.0	no	611.2	548.4
3	10.0	no	587.0	528.1
4	15.0	no	580.9	528.8
5	20.0	no	602.7	523.6
	Average		324.4	536.8
	Median		324.4	536.8
	Std. Dev.		N/A	N/A



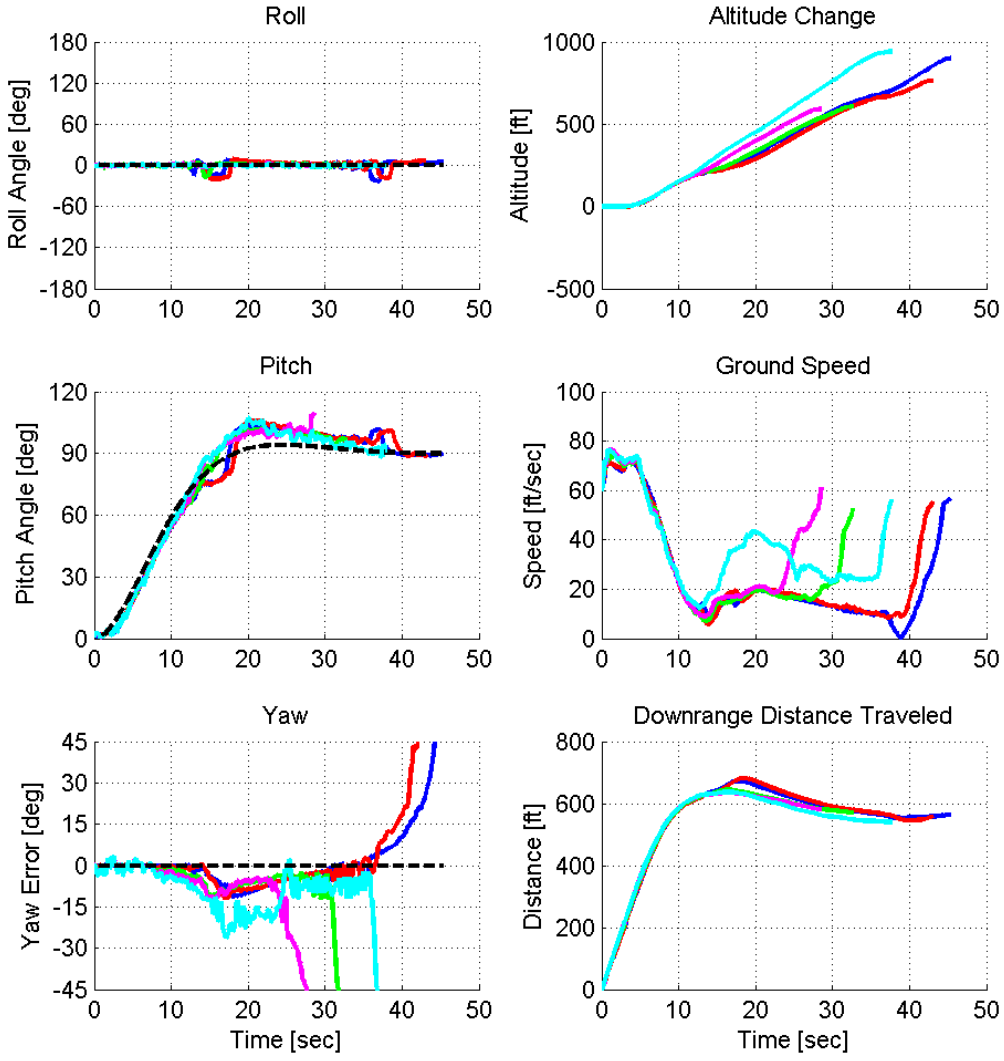
Simulation: PID Reference Model Response, Rise Time = 10 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	740.2	669.1
2	5.0	no	765.9	678.5
3	10.0	no	572.7	636.0
4	15.0	no	584.5	623.9
5	20.0	no	524.5	632.6
	Average		740.2	669.1
	Median		740.2	669.1
	Std. Dev.		N/A	N/A



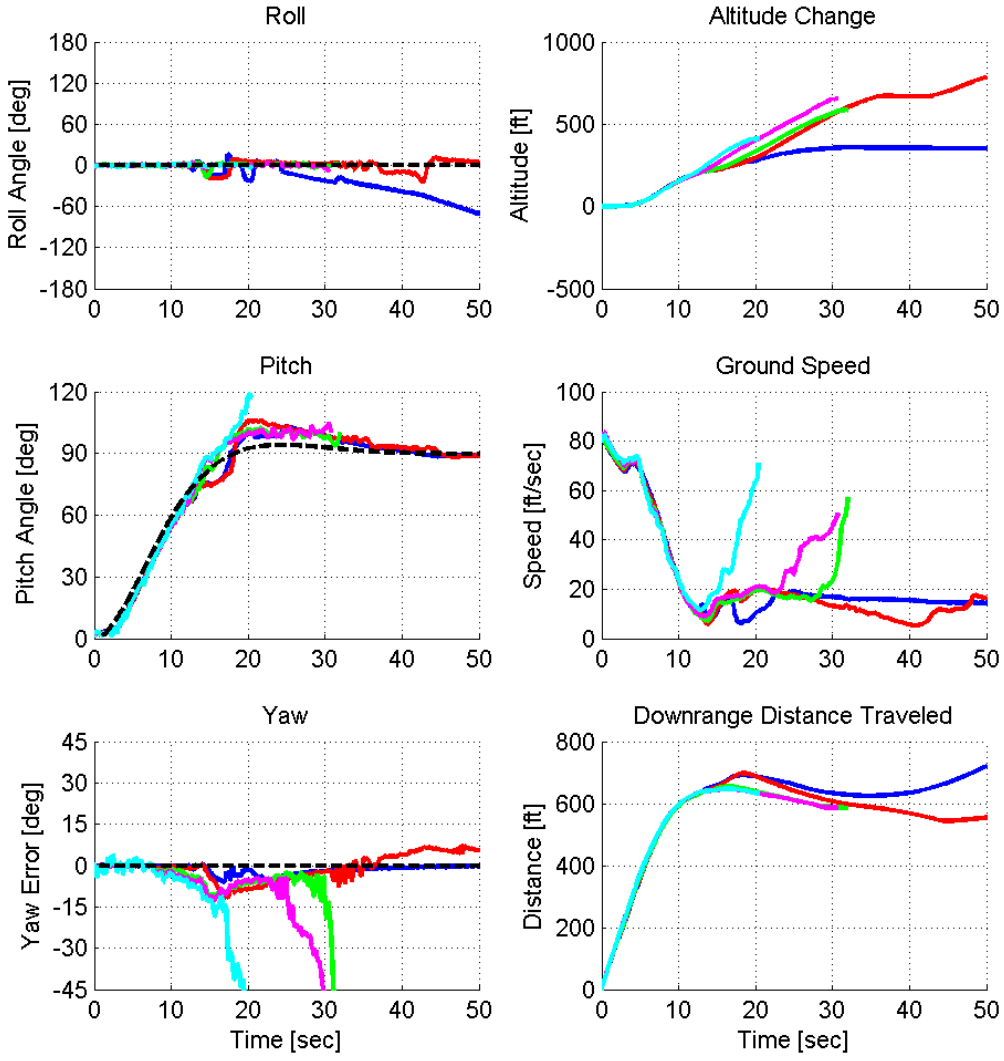
Simulation: PID Reference Model Response, Rise Time = 10 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	900.0	672.8
2	5.0	no	765.6	682.0
3	10.0	no	608.1	643.4
4	15.0	no	590.9	634.7
5	20.0	no	942.1	637.3
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



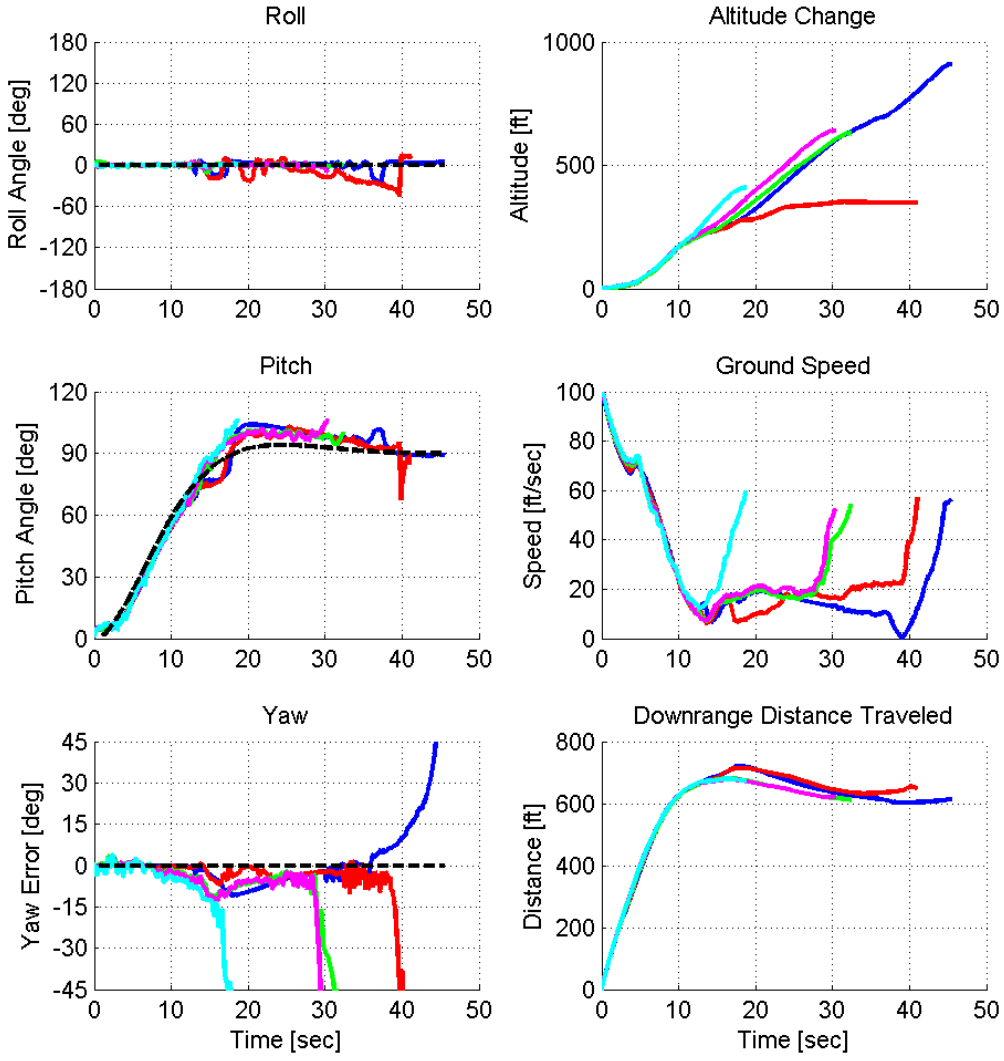
Simulation: PID Reference Model Response, Rise Time = 10 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	359.5	721.3
2	5.0	yes	783.8	697.7
3	10.0	no	586.4	654.9
4	15.0	no	655.6	646.3
5	20.0	no	411.5	647.4
	Average		571.6	709.5
	Median		571.6	709.5
	Std. Dev.		300.0	16.7



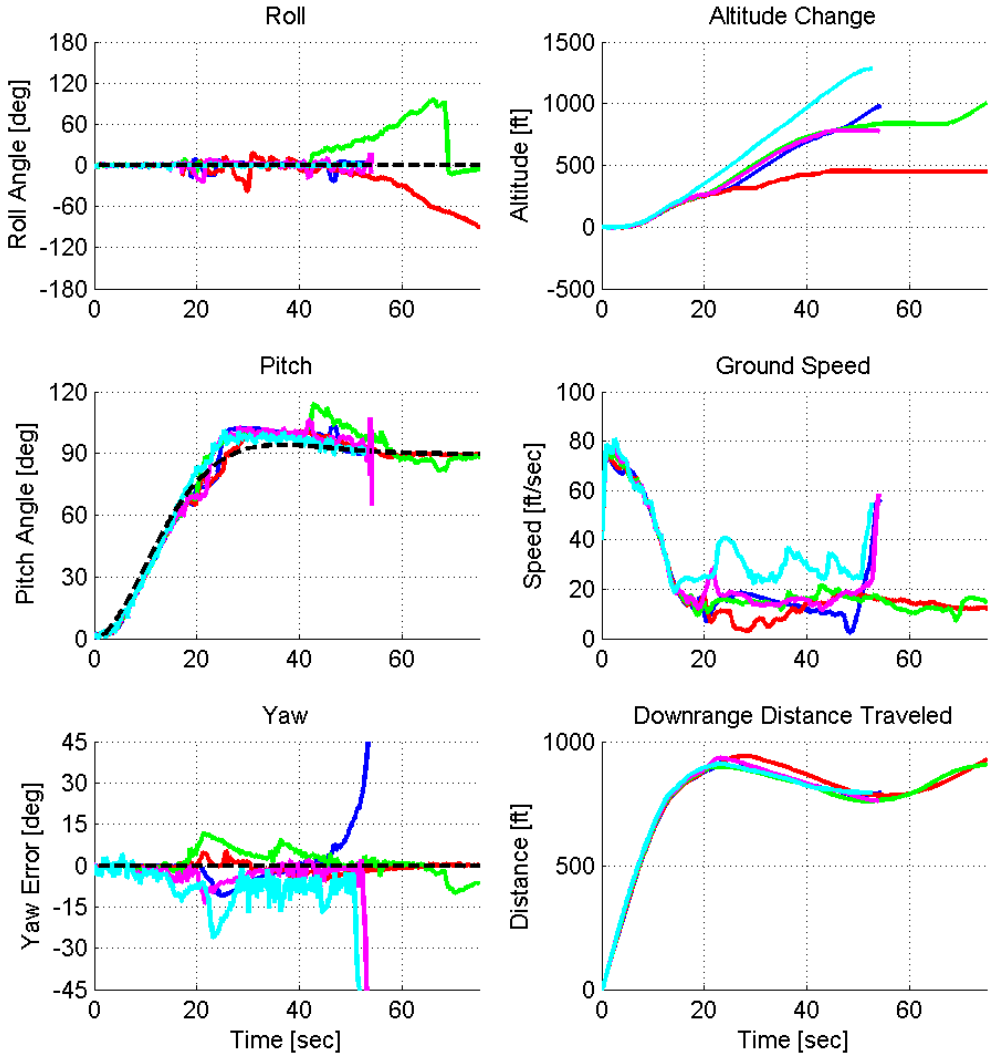
Simulation: PID Reference Model Response, Rise Time = 10 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	910.6	721.4
2	5.0	no	352.1	714.2
3	10.0	no	630.2	681.6
4	15.0	no	639.6	677.9
5	20.0	no	409.7	679.5
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



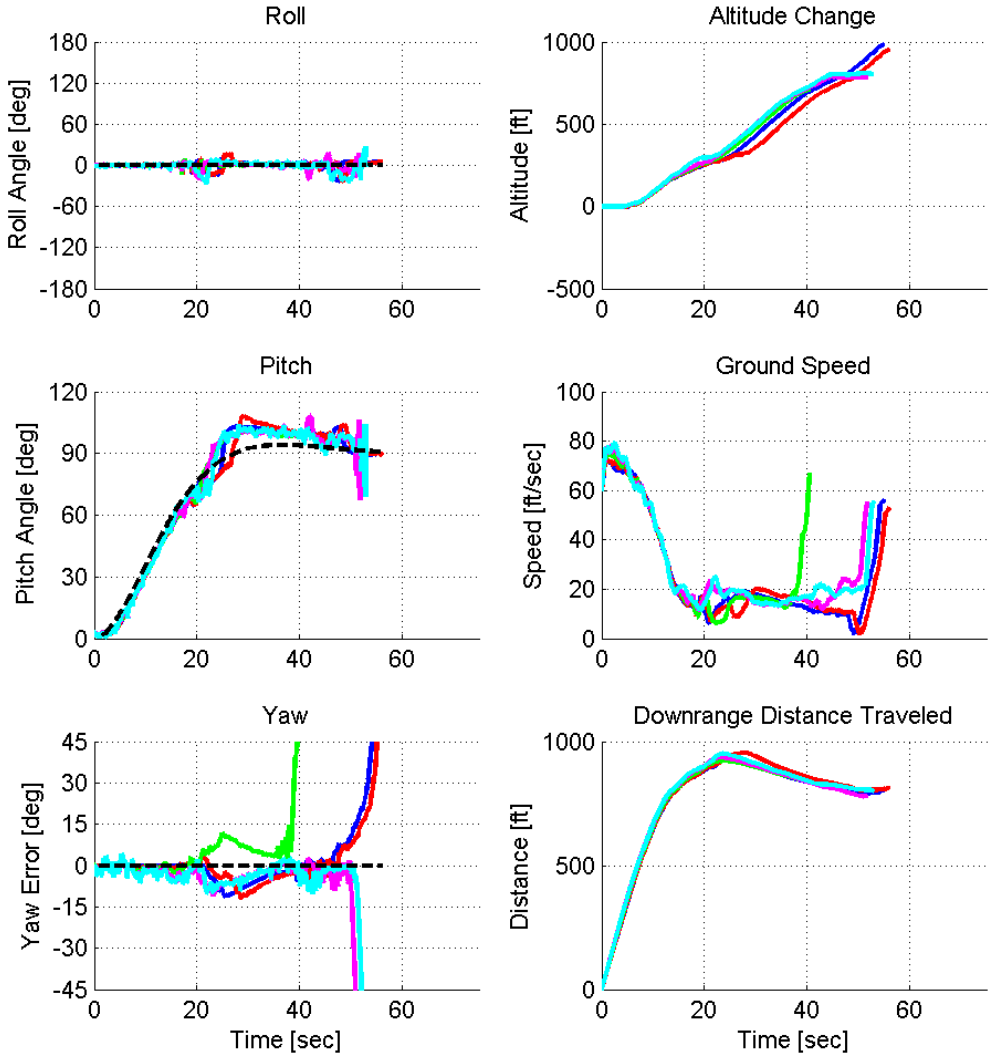
Simulation: PID Reference Model Response, Rise Time = 15 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	979.2	927.5
2	5.0	yes	454.9	941.2
3	10.0	yes	1000.8	908.4
4	15.0	no	782.2	933.3
5	20.0	no	1281.5	910.4
	Average		727.9	924.8
	Median		727.9	924.8
	Std. Dev.		386.0	23.2



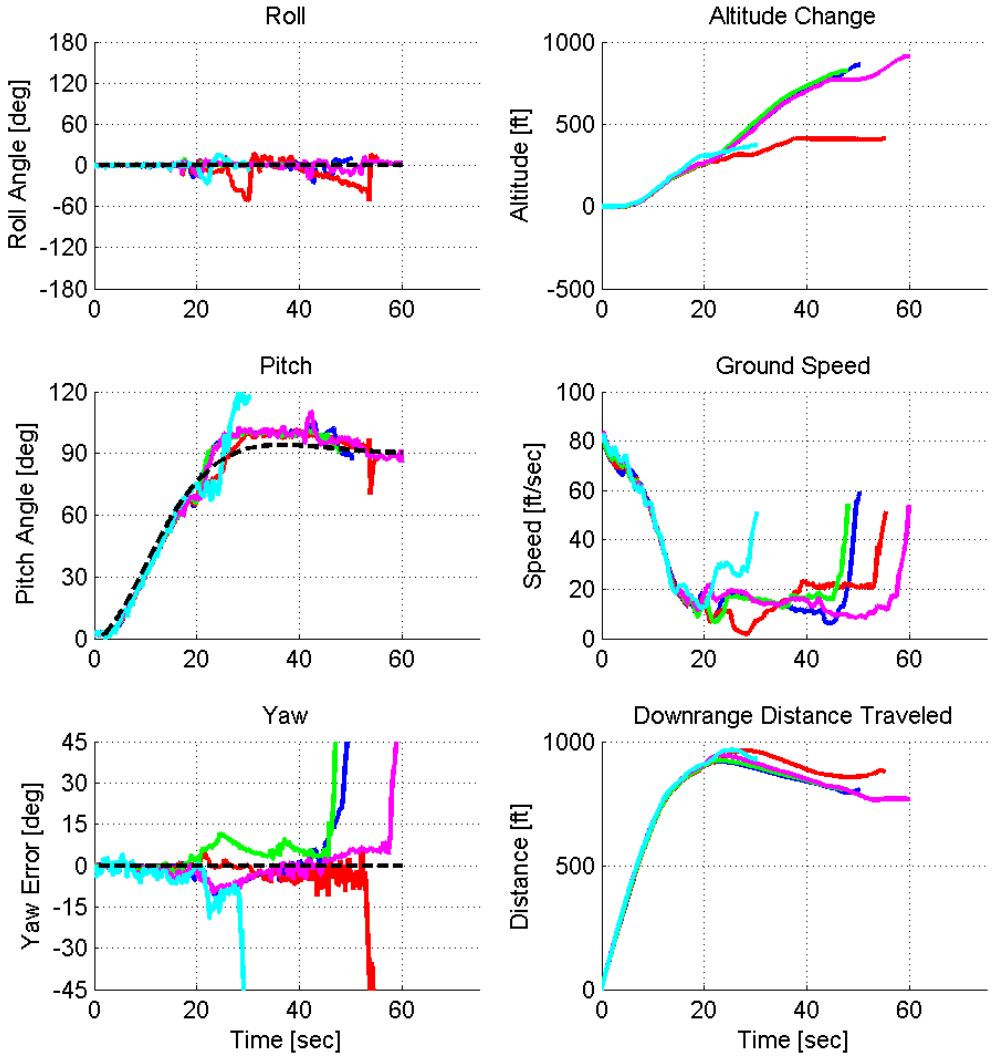
Simulation: PID Reference Model Response, Rise Time = 15 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	983.2	937.6
2	5.0	no	952.1	954.5
3	10.0	no	708.1	922.6
4	15.0	no	788.2	936.0
5	20.0	no	809.0	951.0
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



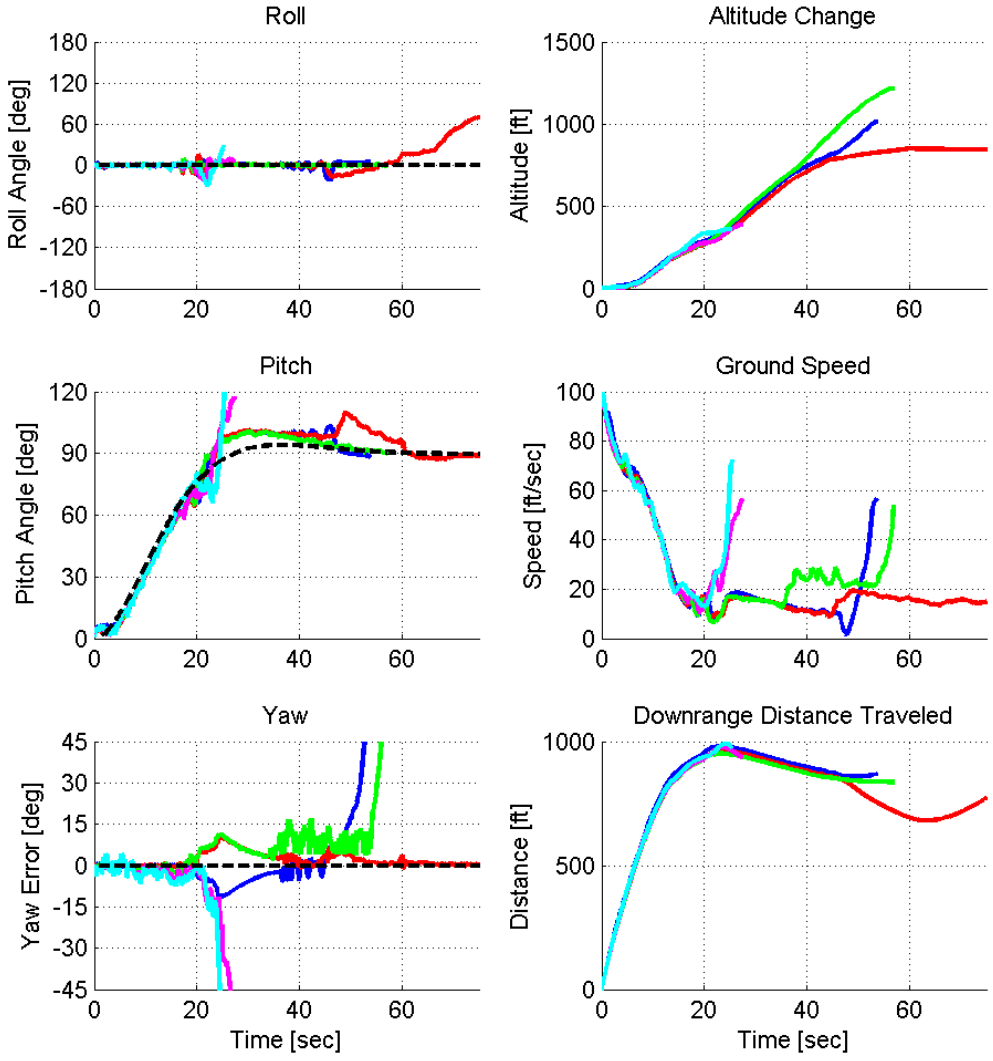
Simulation: PID Reference Model Response, Rise Time = 15 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	859.5	919.9
2	5.0	no	413.2	962.8
3	10.0	no	823.6	924.2
4	15.0	no	910.9	945.2
5	20.0	no	372.5	965.1
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



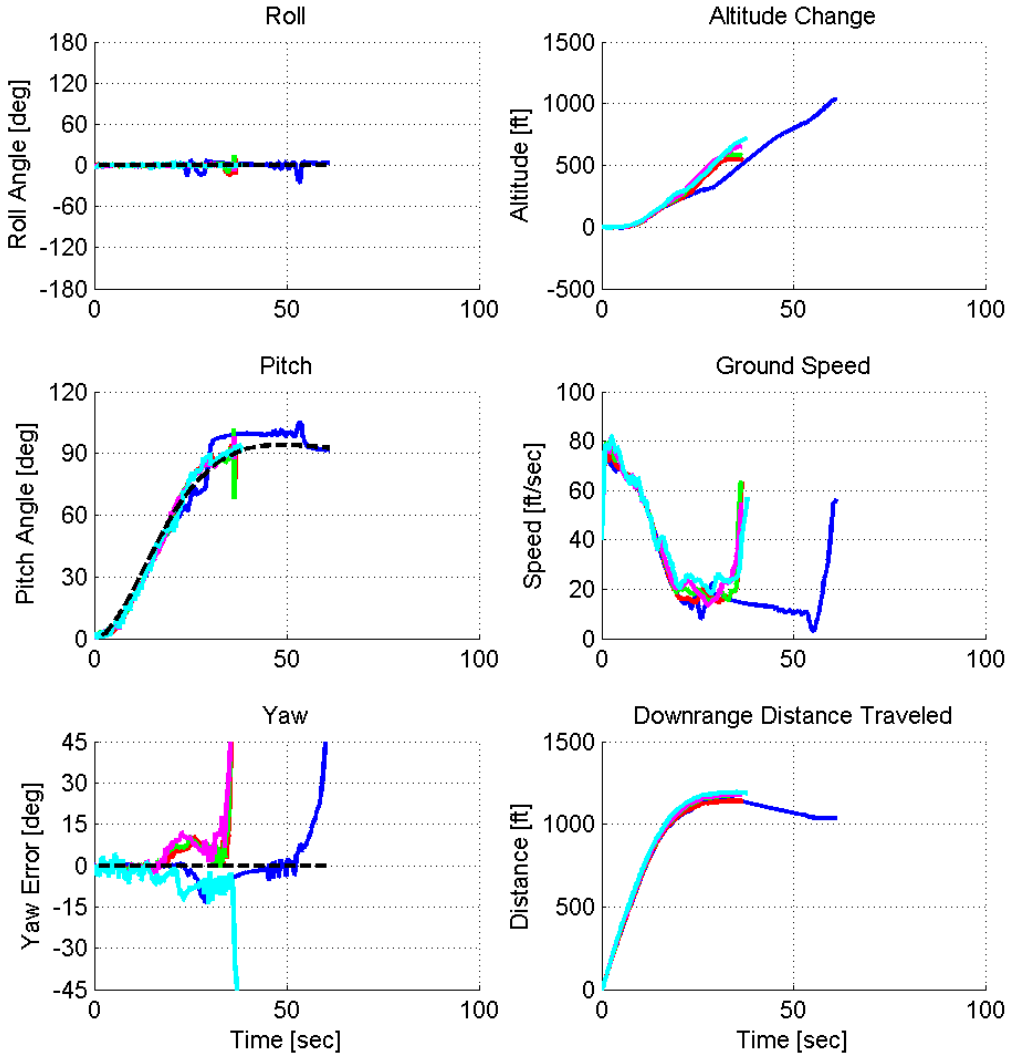
Simulation: PID Reference Model Response, Rise Time = 15 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	1014.6	982.0
2	5.0	yes	851.7	964.4
3	10.0	no	1217.3	951.0
4	15.0	no	388.7	977.0
5	20.0	no	362.5	992.1
	Average		851.7	964.4
	Median		851.7	964.4
	Std. Dev.		N/A	N/A



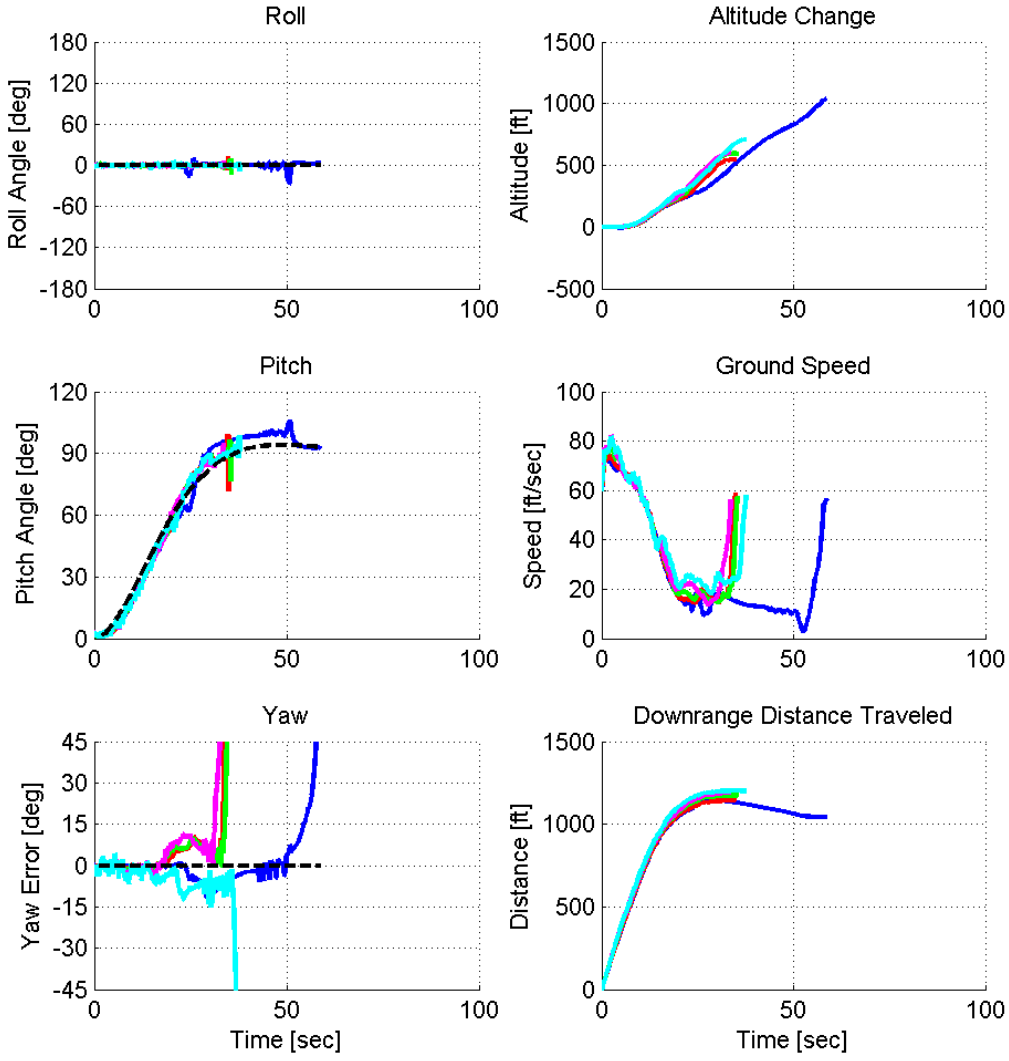
Simulation: PID Reference Model Response, Rise Time = 20 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	1035.8	1168.1
2	5.0	no	550.0	1141.4
3	10.0	no	586.1	1180.3
4	15.0	no	653.1	1178.7
5	20.0	no	717.2	1192.0
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



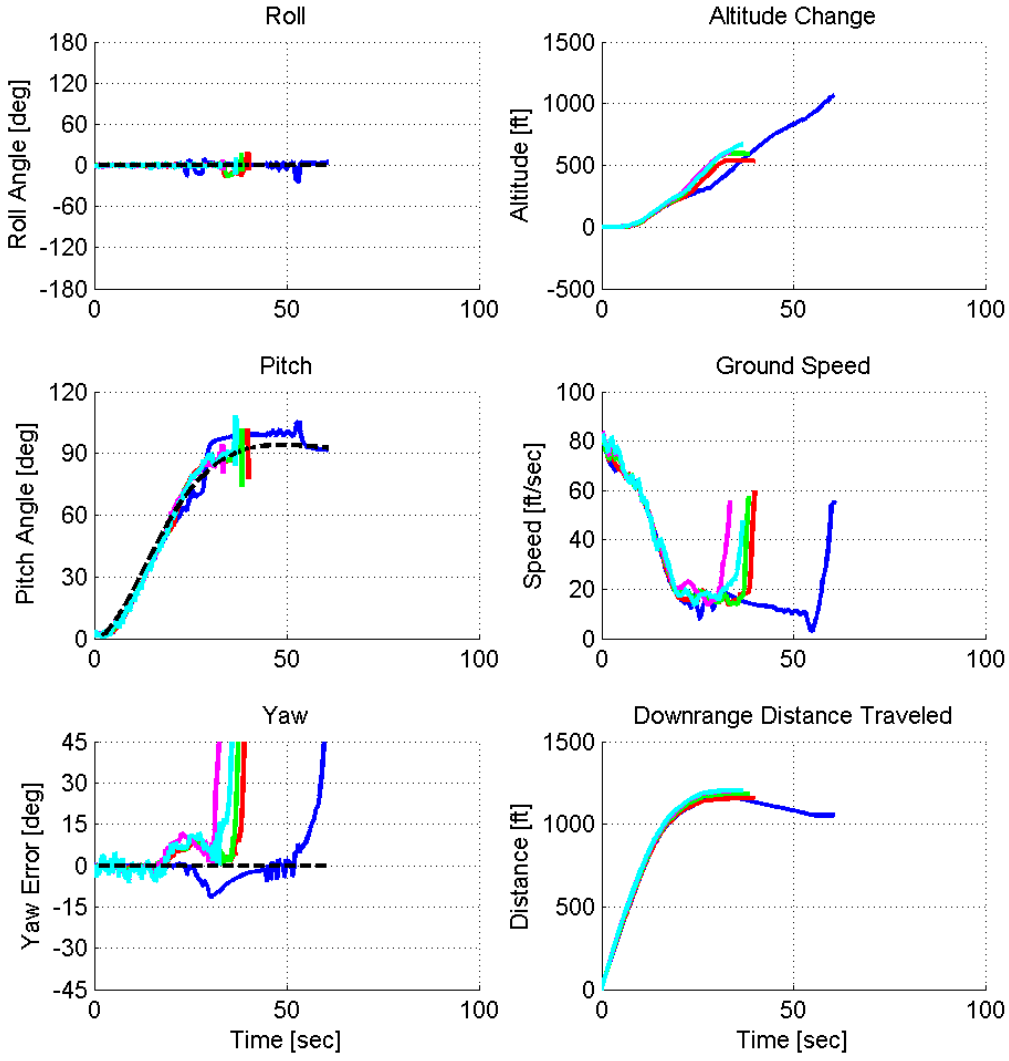
Simulation: PID Reference Model Response, Rise Time = 20 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	1039.1	1145.5
2	5.0	no	547.7	1145.4
3	10.0	no	597.6	1175.5
4	15.0	no	602.5	1189.8
5	20.0	no	708.6	1202.9
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



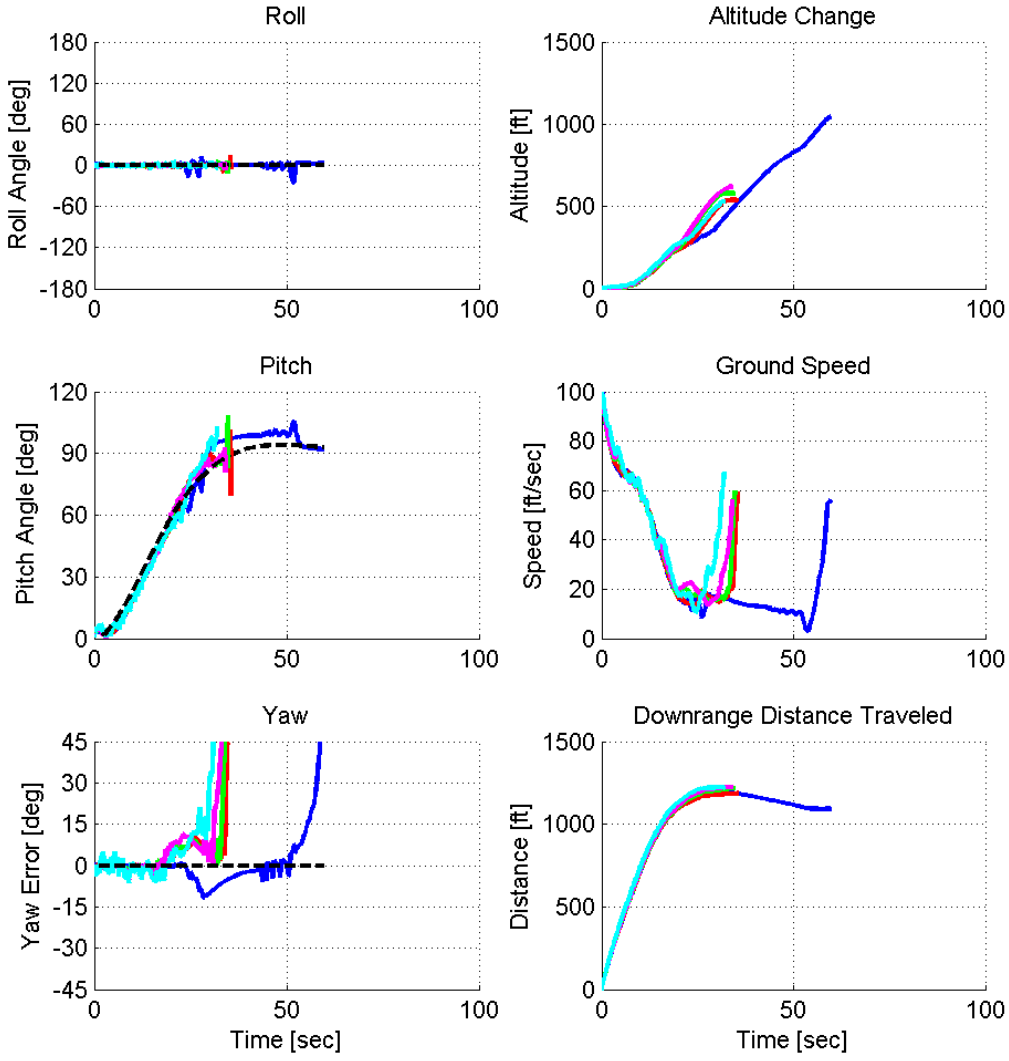
Simulation: PID Reference Model Response, Rise Time = 20 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	1066.2	1180.0
2	5.0	no	540.1	1160.9
3	10.0	no	596.5	1186.7
4	15.0	no	600.7	1201.7
5	20.0	no	668.2	1204.6
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



Simulation: PID Reference Model Response, Rise Time = 20 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	1045.4	1203.6
2	5.0	no	540.0	1186.4
3	10.0	no	581.7	1218.5
4	15.0	no	621.0	1223.8
5	20.0	no	530.2	1223.7
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



A.3 Model Reference Adaptive Control

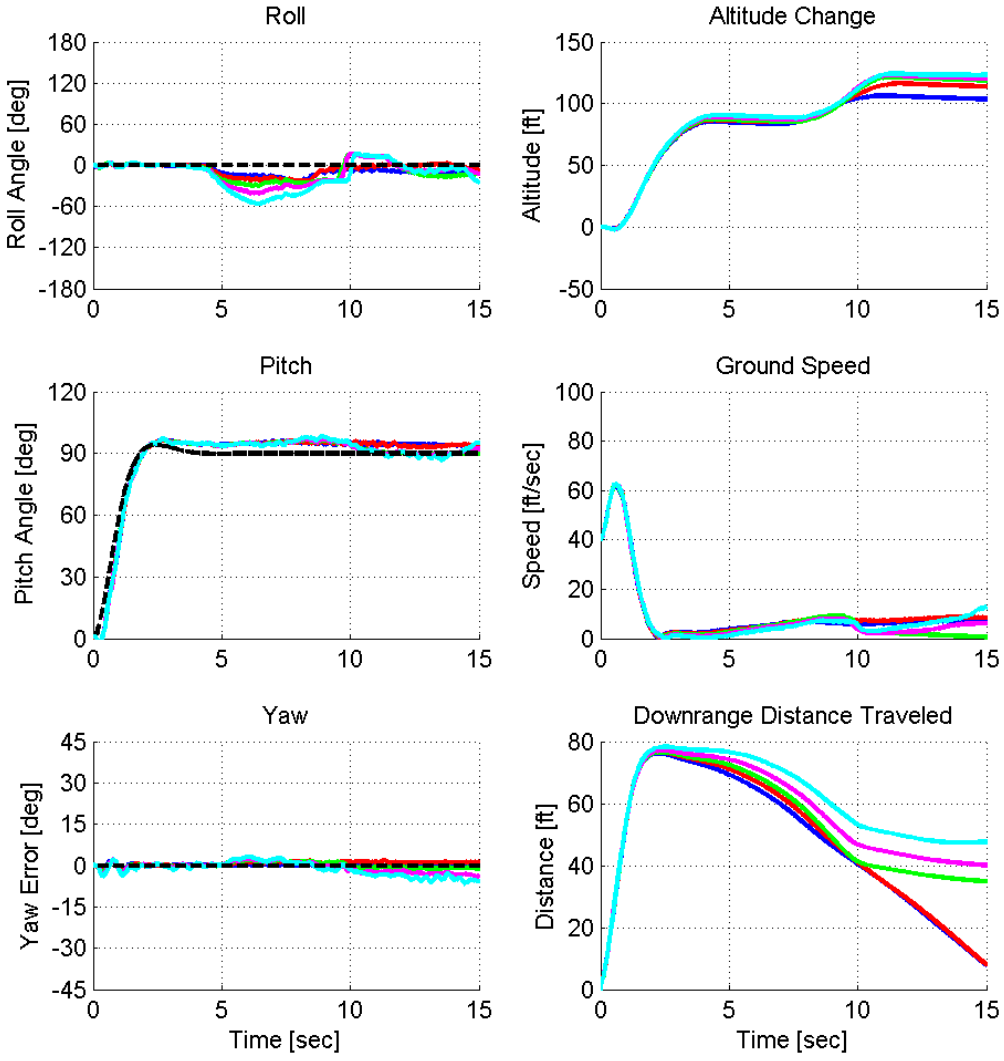
MRAC Simulations: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
5	80%	60%	80%	100%	80%
7	40%	20%	40%	0%	25%
10	40%	20%	40%	20%	30%
15	20%	0%	0%	0%	5%
20	0%	20%	0%	0%	5%
Cumulative	60%	53%	58%	53%	56%

MRAC Simulations				
Rise Time	Approach Speed	Max. Altitude Change [ft]		
		Average	Median	Std. Dev.
1 sec	40 ft/sec	118.4	121.6	7.3
	60 ft/sec	140.2	140.8	6.5
	80 ft/sec	157.5	158.3	2.2
	100 ft/sec	174.2	175.8	3.8
2 sec	40 ft/sec	152.2	152.4	10.2
	60 ft/sec	164.9	167.2	6.7
	80 ft/sec	186.9	190.5	10.2
	100 ft/sec	206.8	209.0	6.8
3 sec	40 ft/sec	166.7	166.1	7.0
	60 ft/sec	183.0	184.9	7.6
	80 ft/sec	198.2	202.5	8.7
	100 ft/sec	212.6	214.5	6.1
5 sec	40 ft/sec	189.5	187.7	4.1
	60 ft/sec	202.6	202.5	2.1
	80 ft/sec	200.8	206.4	14.9
	100 ft/sec	211.1	217.9	16.5
7 sec	40 ft/sec	175.0	175.0	35.7
	60 ft/sec	167.6	167.6	N/A
	80 ft/sec	196.0	196.0	29.5
	100 ft/sec	N/A	N/A	N/A
10 sec	40 ft/sec	219.0	219.0	43.3
	60 ft/sec	189.8	189.8	N/A
	80 ft/sec	212.8	212.8	13.7
	100 ft/sec	213.2	213.2	N/A
15 sec	40 ft/sec	356.2	356.2	N/A
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	N/A	N/A	N/A
20 sec	40 ft/sec	N/A	N/A	N/A
	60 ft/sec	215.1	215.1	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	N/A	N/A	N/A

MRAC Simulations				
Rise Time	Approach Speed	Max. Distance Traveled [ft]		
		Average	Median	Std. Dev.
1 sec	40 ft/sec	76.9	76.9	0.8
	60 ft/sec	92.5	92.3	0.7
	80 ft/sec	106.3	106.2	0.7
	100 ft/sec	120.6	120.5	0.4
2 sec	40 ft/sec	153.4	153.2	2.8
	60 ft/sec	175.0	174.7	2.7
	80 ft/sec	197.2	196.9	2.3
	100 ft/sec	220.4	220.2	2.2
3 sec	40 ft/sec	231.0	230.6	1.5
	60 ft/sec	257.7	258.2	1.6
	80 ft/sec	284.7	284.8	2.1
	100 ft/sec	311.4	310.6	2.1
5 sec	40 ft/sec	376.3	376.0	2.3
	60 ft/sec	404.8	406.0	2.3
	80 ft/sec	431.5	432.1	3.0
	100 ft/sec	465.3	465.0	3.5
7 sec	40 ft/sec	501.5	501.6	5.1
	60 ft/sec	532.7	532.7	N/A
	80 ft/sec	563.0	563.0	5.9
	100 ft/sec	N/A	N/A	N/A
10 sec	40 ft/sec	858.5	858.5	65.2
	60 ft/sec	809.9	809.9	N/A
	80 ft/sec	824.9	824.9	66.7
	100 ft/sec	798.3	798.3	N/A
15 sec	40 ft/sec	1008.2	1008.2	N/A
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	N/A	N/A	N/A
20 sec	40 ft/sec	N/A	N/A	N/A
	60 ft/sec	1822.9	1822.9	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	N/A	N/A	N/A

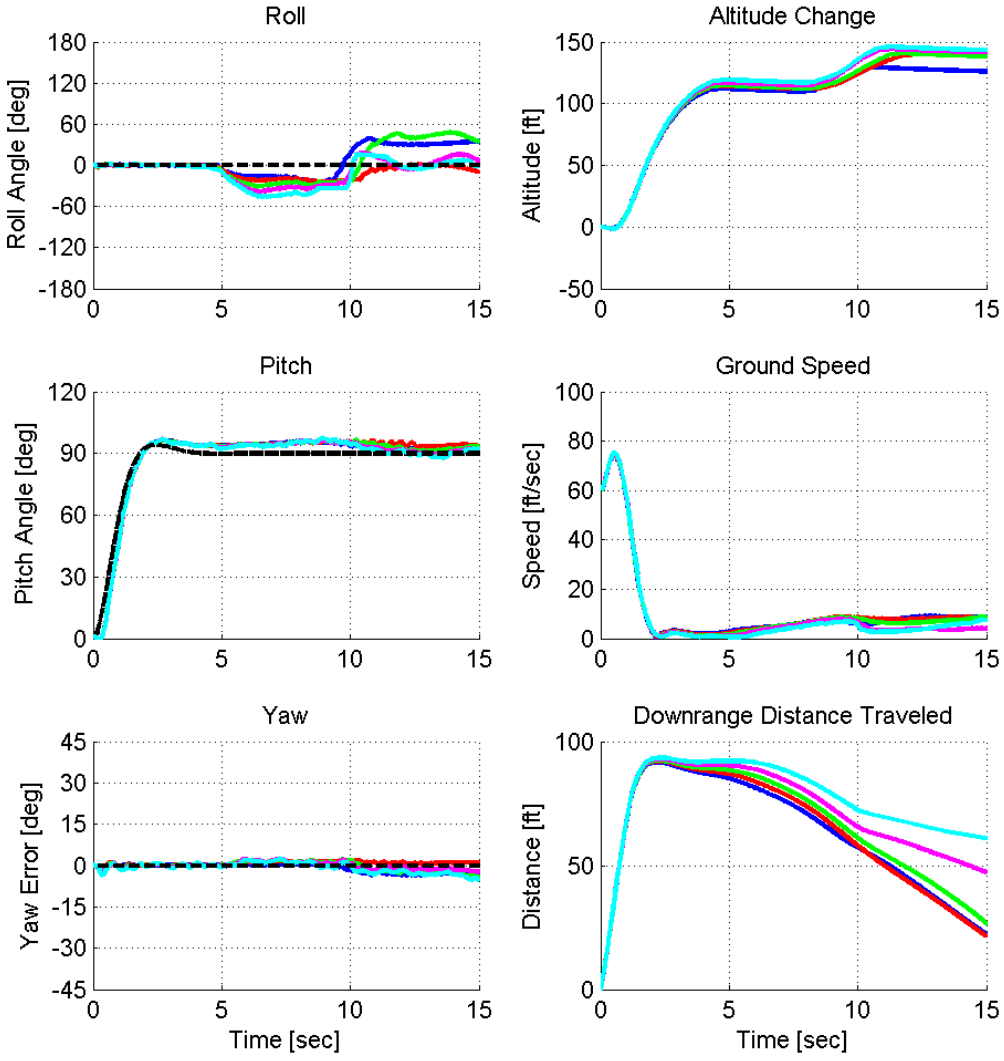
Simulation: MRAC, Rise Time = 1 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	106.5	76.1
2	5.0	yes	116.4	76.5
3	10.0	yes	121.6	76.6
4	15.0	yes	123.0	77.1
5	20.0	yes	124.4	78.2
	Average		118.4	76.9
	Median		121.6	76.6
	Std. Dev.		7.3	0.8



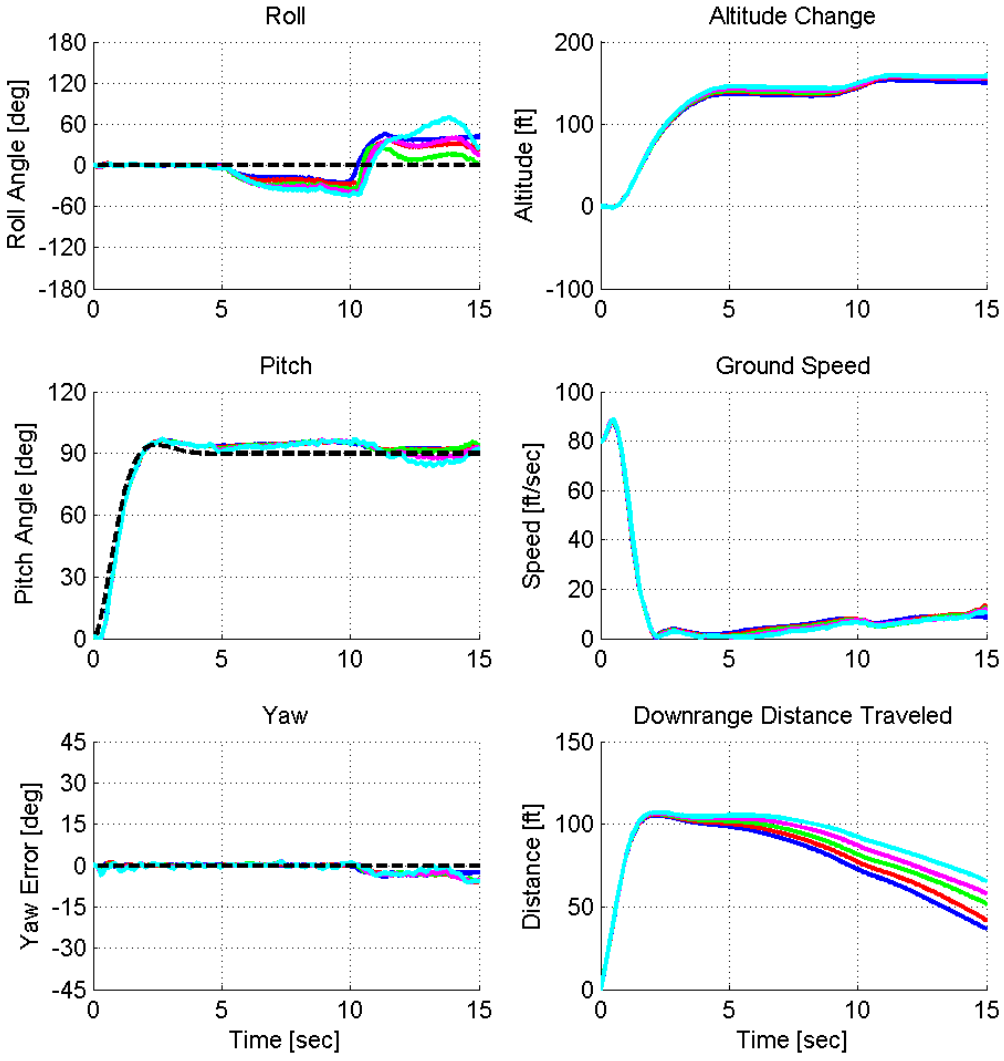
Simulation: MRAC, Rise Time = 1 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	129.3	91.6
2	5.0	yes	140.4	92.1
3	10.0	yes	140.8	92.3
4	15.0	yes	144.3	92.9
5	20.0	yes	146.0	93.4
	Average		140.2	92.5
	Median		140.8	92.3
	Std. Dev.		6.5	0.7



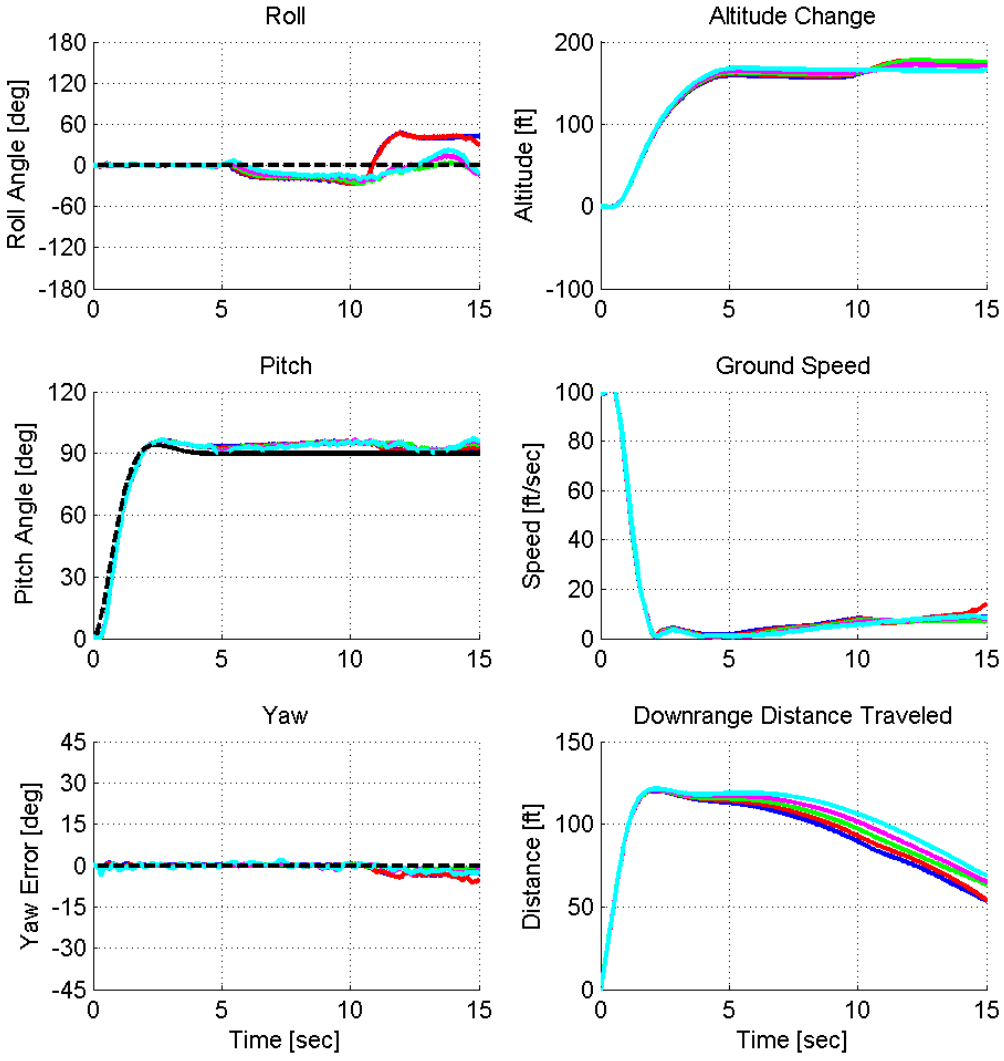
Simulation: MRAC, Rise Time = 1 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	153.7	105.4
2	5.0	yes	157.4	105.9
3	10.0	yes	158.8	106.2
4	15.0	yes	158.3	106.7
5	20.0	yes	159.2	107.1
	Average		157.5	106.3
	Median		158.3	106.2
	Std. Dev.		2.2	0.7



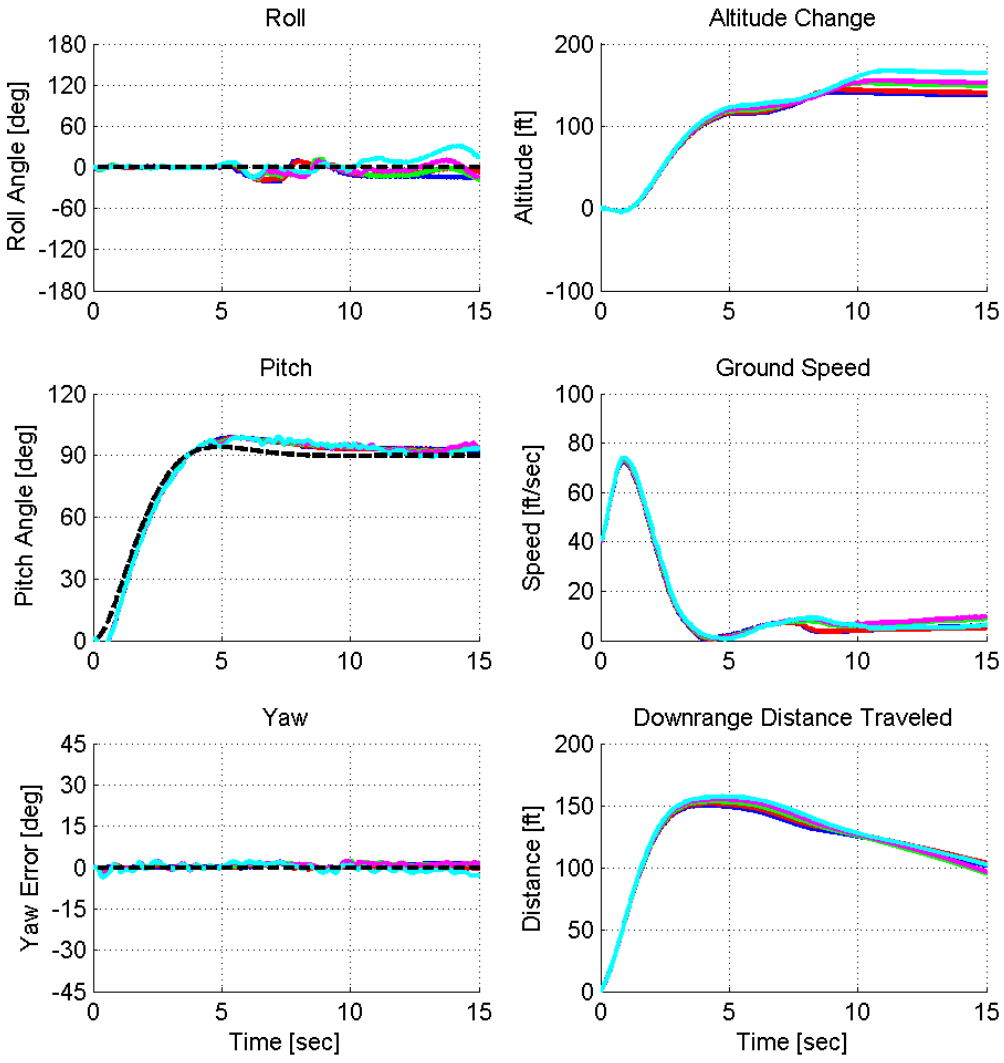
Simulation: MRAC, Rise Time = 1 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	175.8	120.2
2	5.0	yes	176.6	120.4
3	10.0	yes	177.7	120.5
4	15.0	yes	172.4	120.8
5	20.0	yes	168.5	121.3
	Average		174.2	120.6
	Median		175.8	120.5
	Std. Dev.		3.8	0.4



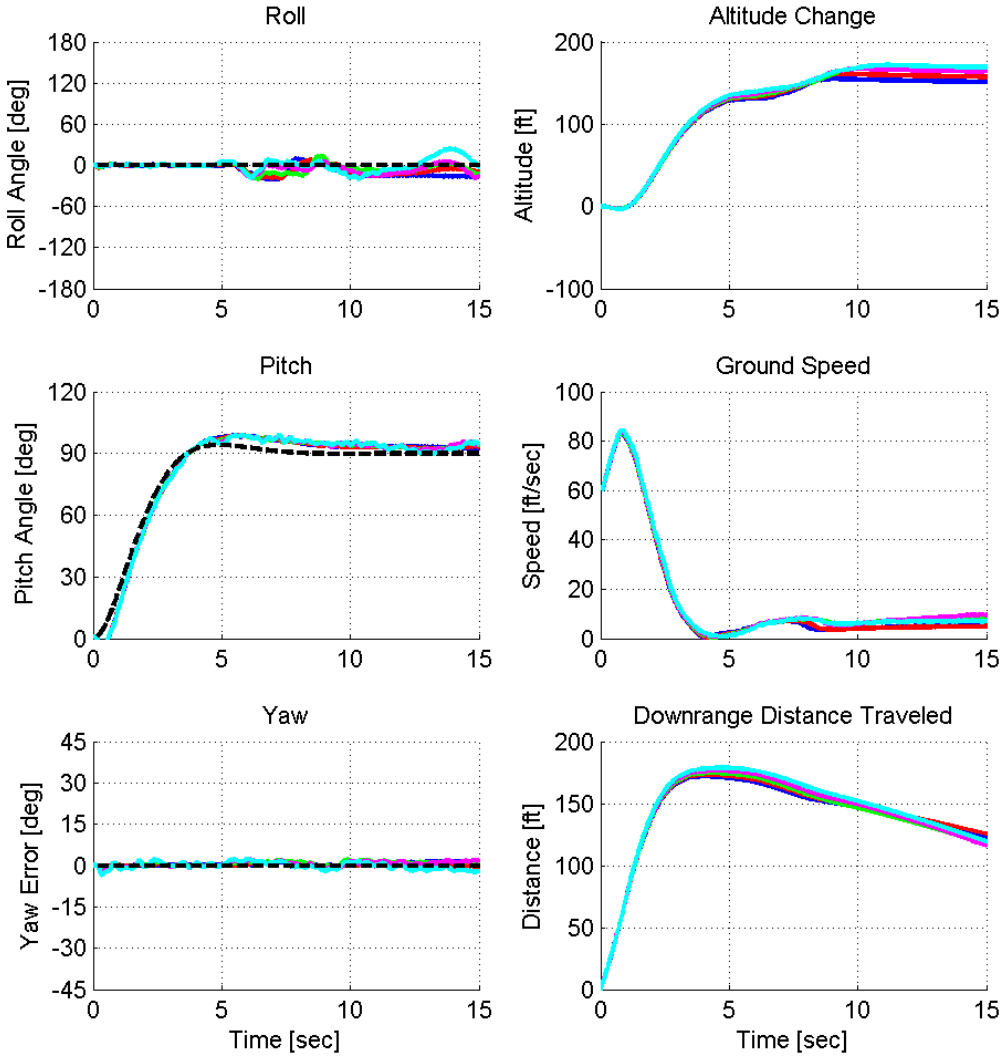
Simulation: MRAC, Rise Time = 2 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	141.4	150.2
2	5.0	yes	144.4	151.5
3	10.0	yes	152.4	153.2
4	15.0	yes	155.6	155.1
5	20.0	yes	167.4	157.1
	Average		152.2	153.4
	Median		152.4	153.2
	Std. Dev.		10.2	2.8



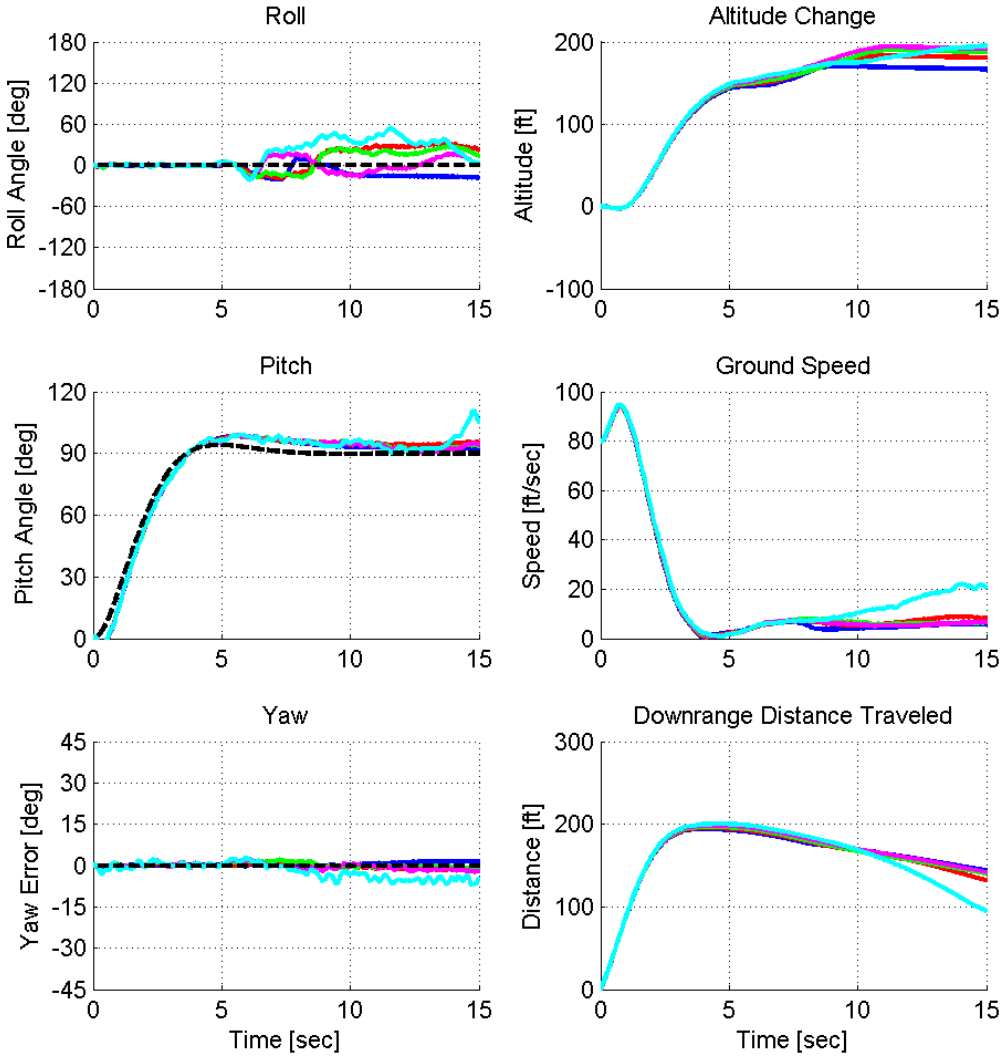
Simulation: MRAC, Rise Time = 2 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	155.1	172.0
2	5.0	yes	161.2	173.3
3	10.0	yes	169.0	174.4
4	15.0	yes	167.2	176.6
5	20.0	yes	171.8	178.7
	Average		164.9	175.0
	Median		167.2	174.7
	Std. Dev.		6.7	2.7



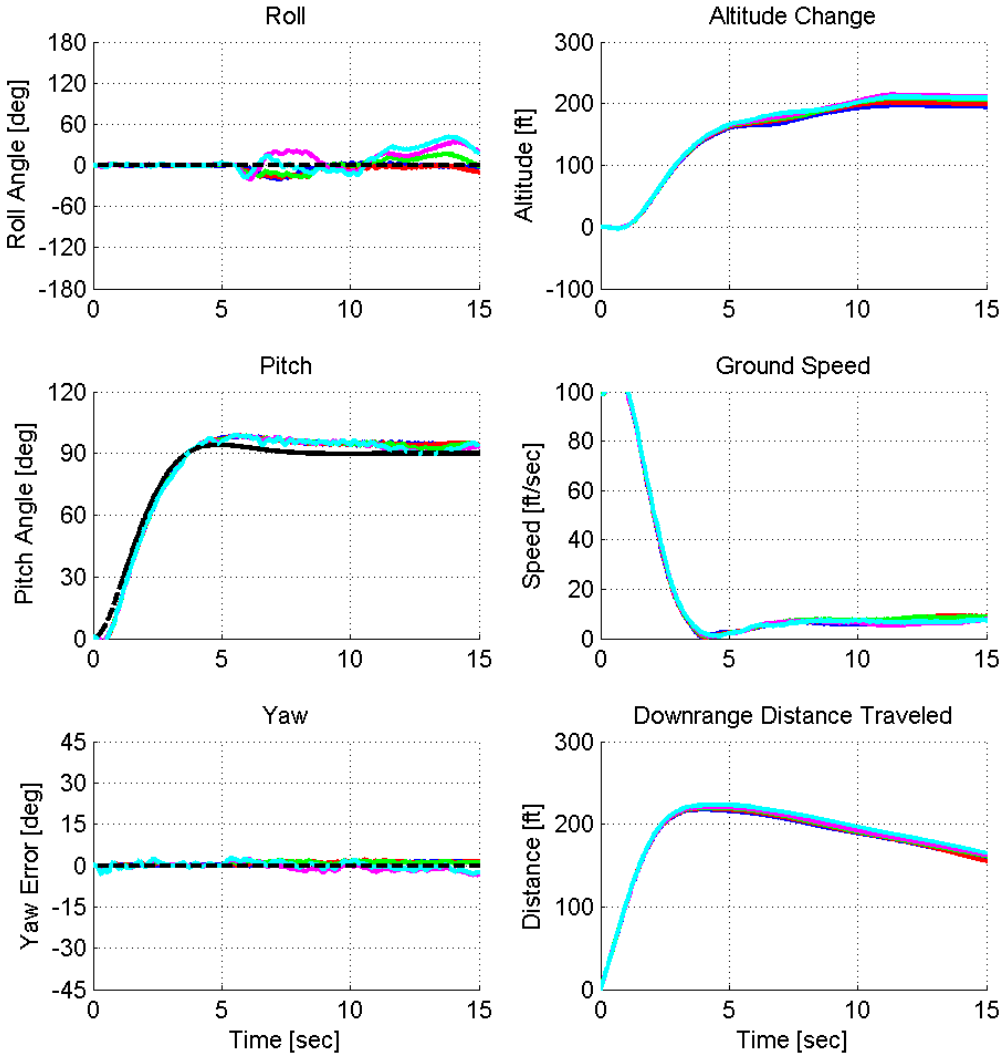
Simulation: MRAC, Rise Time = 2 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	170.8	194.5
2	5.0	yes	183.5	195.7
3	10.0	yes	190.5	196.9
4	15.0	yes	194.4	198.6
5	20.0	yes	195.3	200.4
	Average		186.9	197.2
	Median		190.5	196.9
	Std. Dev.		10.2	2.3



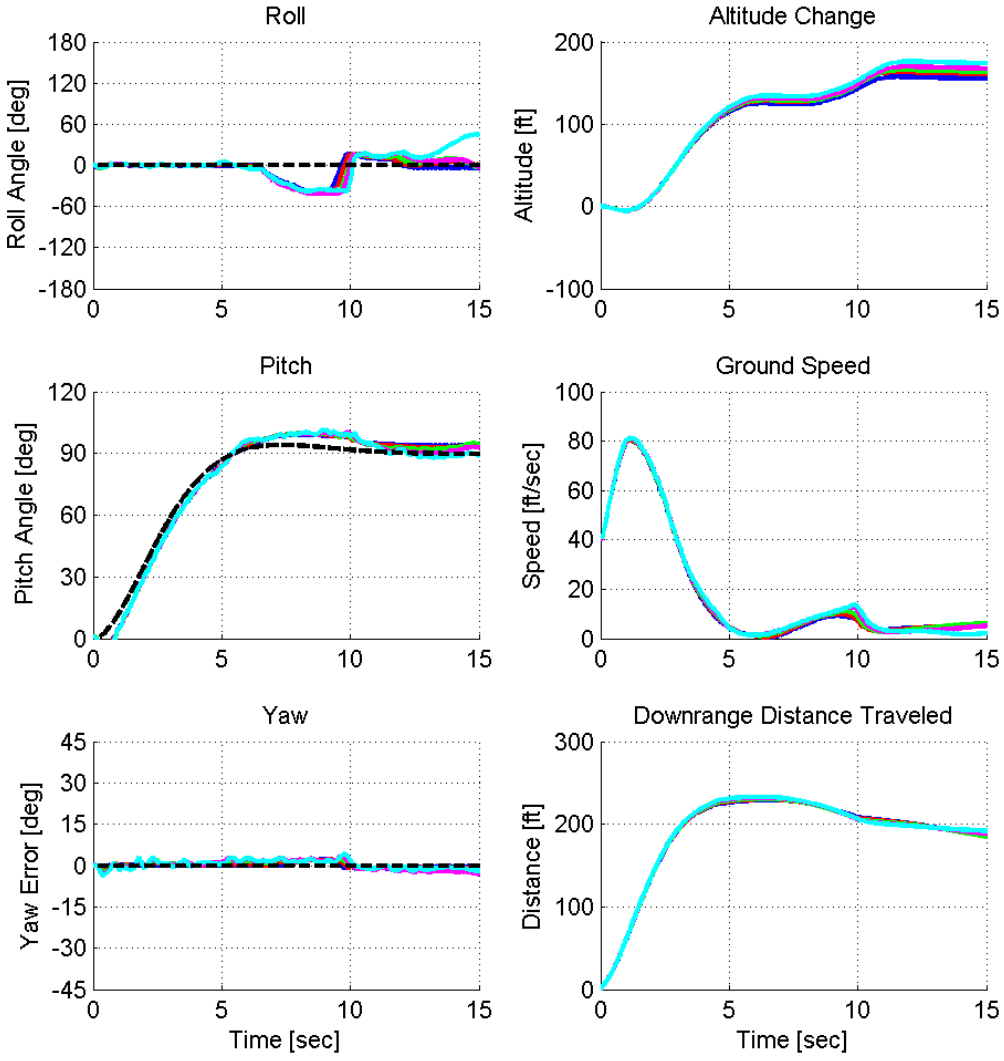
Simulation: MRAC, Rise Time = 2 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	197.4	217.8
2	5.0	yes	202.2	219.1
3	10.0	yes	209.0	220.2
4	15.0	yes	214.0	221.2
5	20.0	yes	211.2	223.6
	Average		206.8	220.4
	Median		209.0	220.2
	Std. Dev.		6.8	2.2



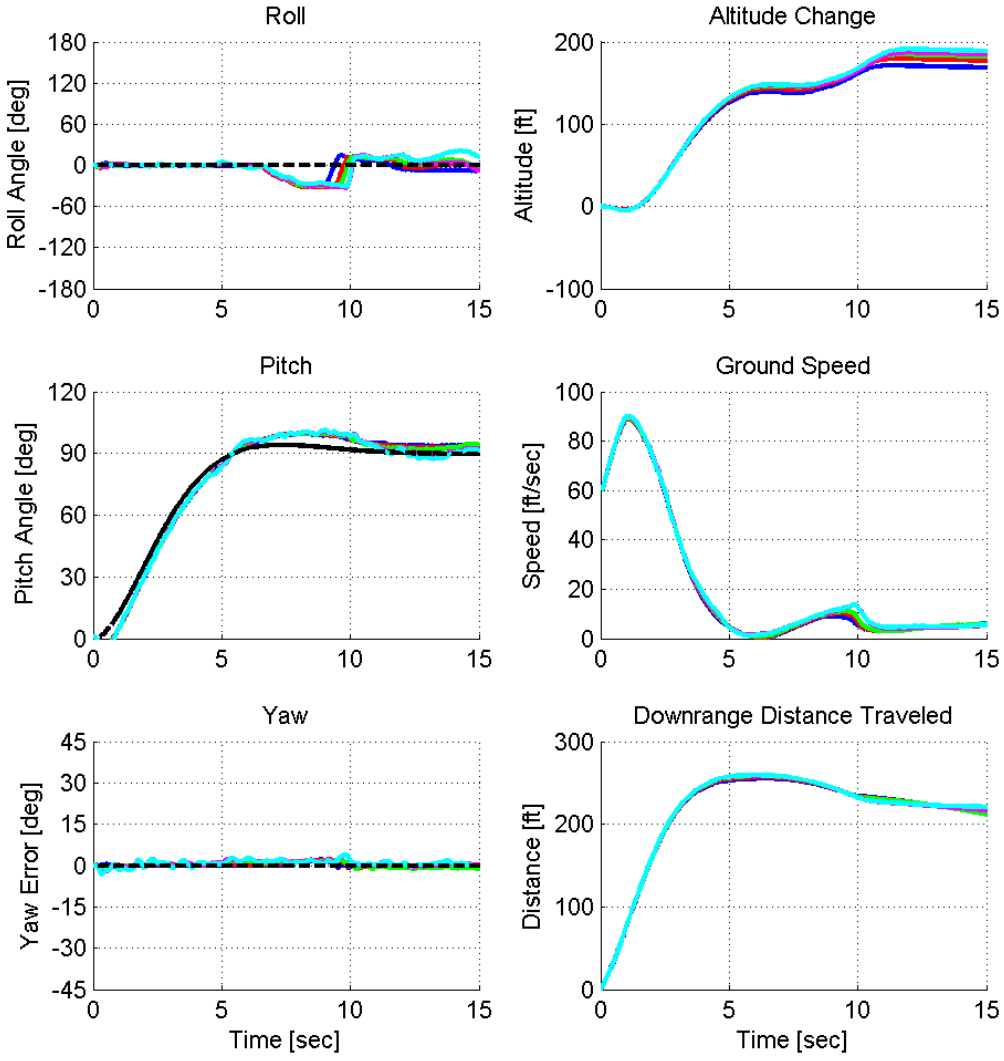
Simulation: MRAC, Rise Time = 3 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	157.7	229.3
2	5.0	yes	163.6	230.0
3	10.0	yes	166.1	230.6
4	15.0	yes	170.0	232.0
5	20.0	yes	176.3	232.9
	Average		166.7	231.0
	Median		166.1	230.6
	Std. Dev.		7.0	1.5



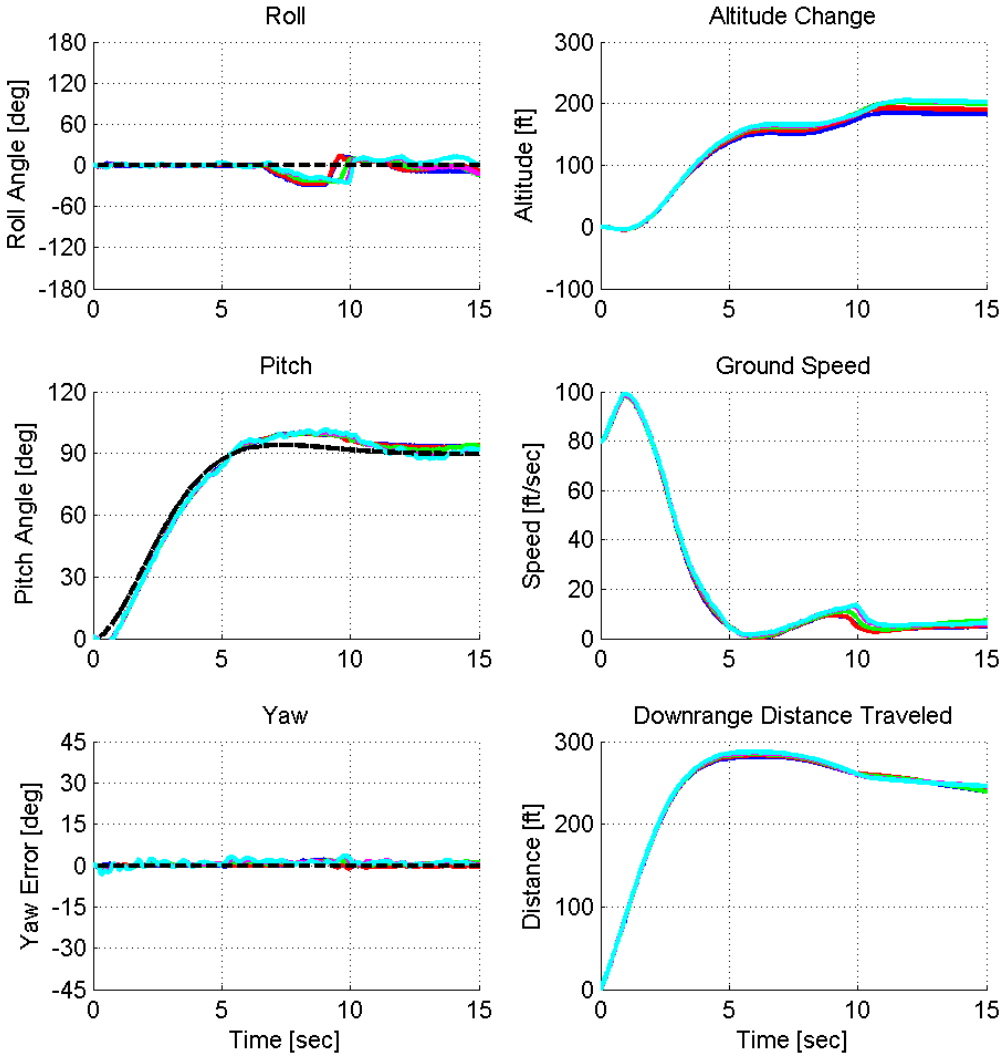
Simulation: MRAC, Rise Time = 3 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	171.6	255.3
2	5.0	yes	180.0	256.7
3	10.0	yes	184.9	258.2
4	15.0	yes	187.4	258.9
5	20.0	yes	191.3	259.2
	Average		183.0	257.7
	Median		184.9	258.2
	Std. Dev.		7.6	1.6



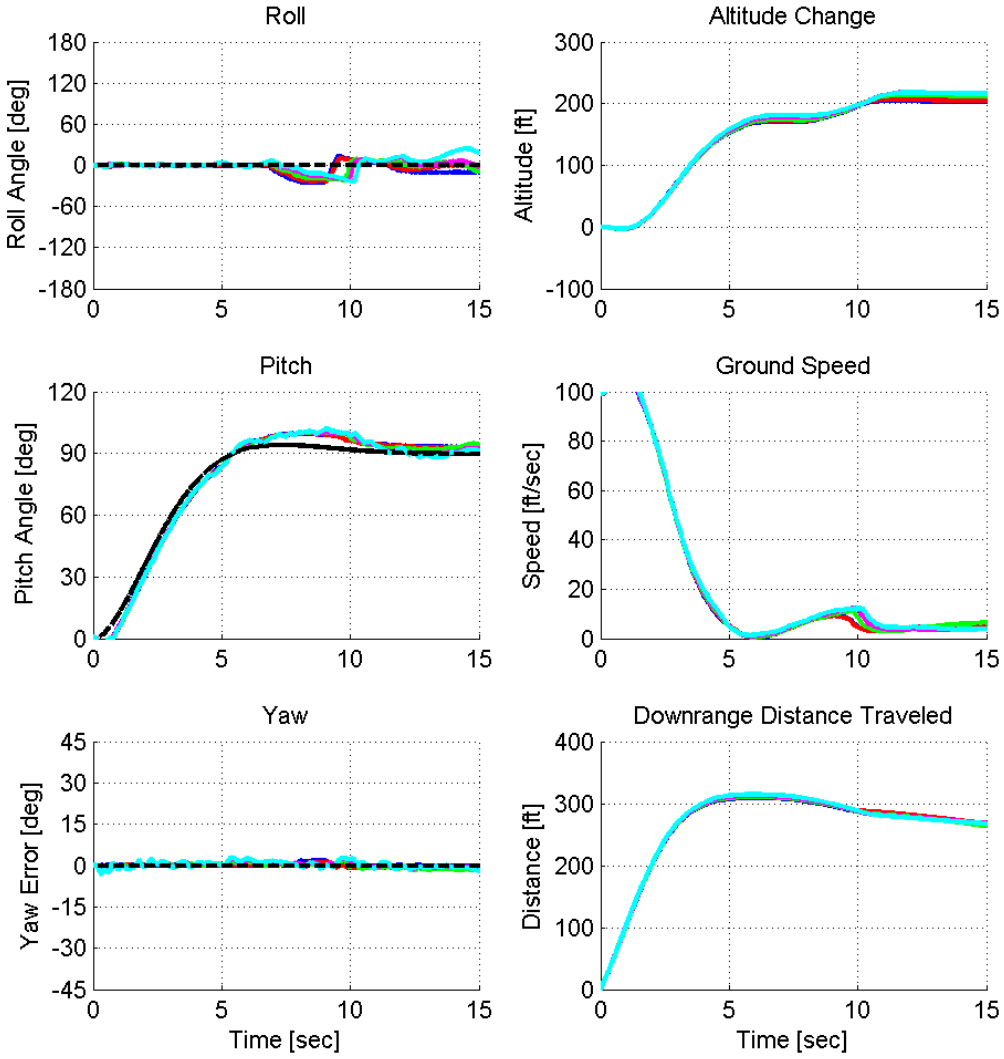
Simulation: MRAC, Rise Time = 3 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	185.4	281.9
2	5.0	yes	192.9	283.4
3	10.0	yes	202.5	284.8
4	15.0	yes	205.0	286.0
5	20.0	yes	205.1	287.3
	Average		198.2	284.7
	Median		202.5	284.8
	Std. Dev.		8.7	2.1



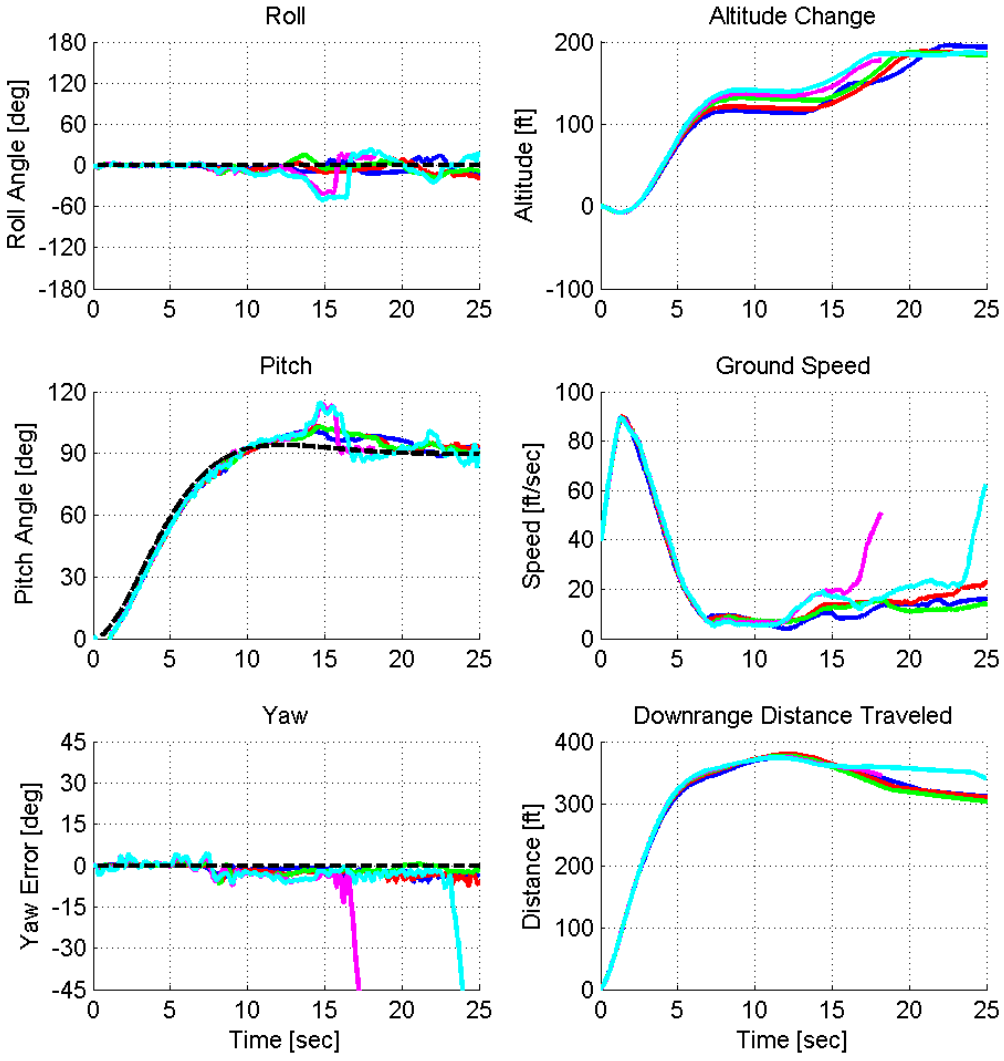
Simulation: MRAC, Rise Time = 3 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	205.1	309.0
2	5.0	yes	207.2	310.4
3	10.0	yes	214.5	310.6
4	15.0	yes	218.2	312.4
5	20.0	yes	218.1	314.4
	Average		212.6	311.4
	Median		214.5	310.6
	Std. Dev.		6.1	2.1



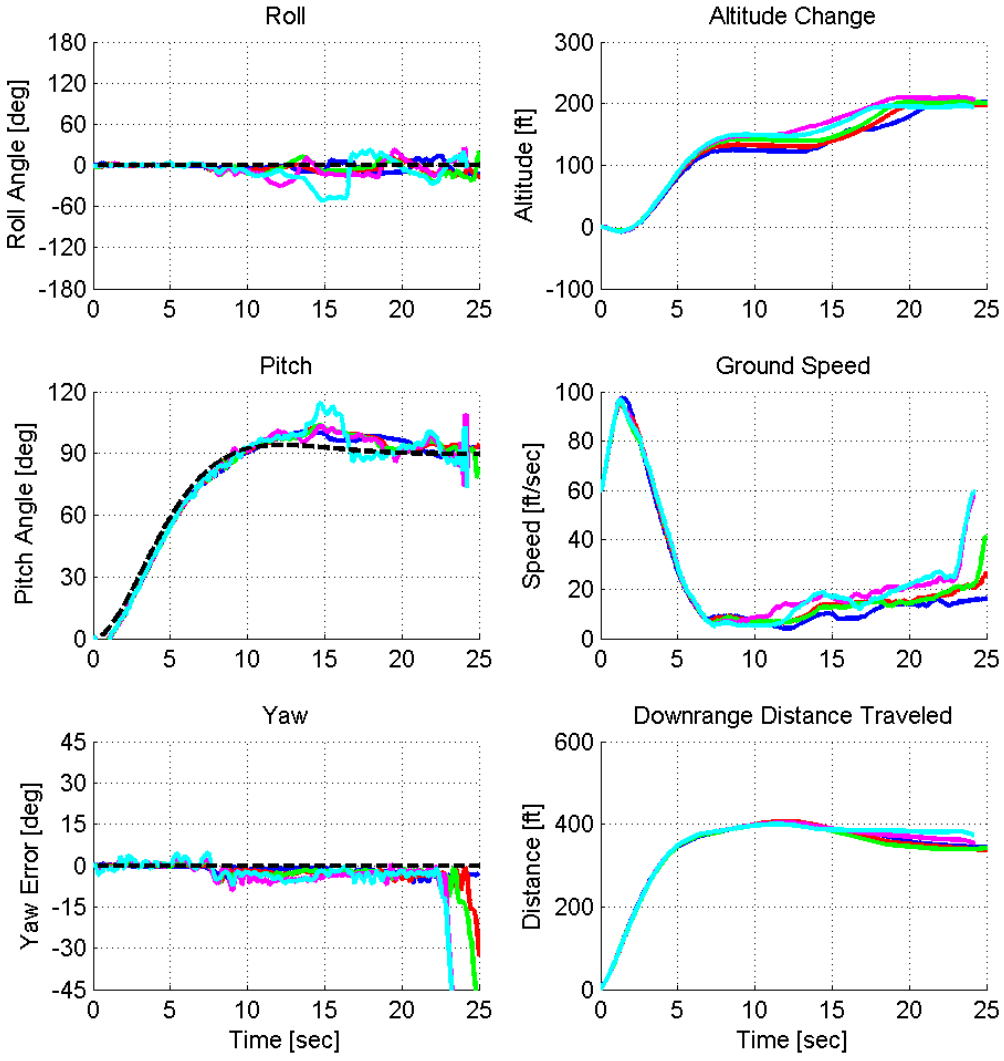
Simulation: MRAC, Rise Time = 5 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	195.5	375.3
2	5.0	yes	188.2	379.3
3	10.0	yes	187.2	376.7
4	15.0	no	177.4	374.7
5	20.0	yes	186.9	373.8
	Average		189.5	376.3
	Median		187.7	376.0
	Std. Dev.		4.1	2.3



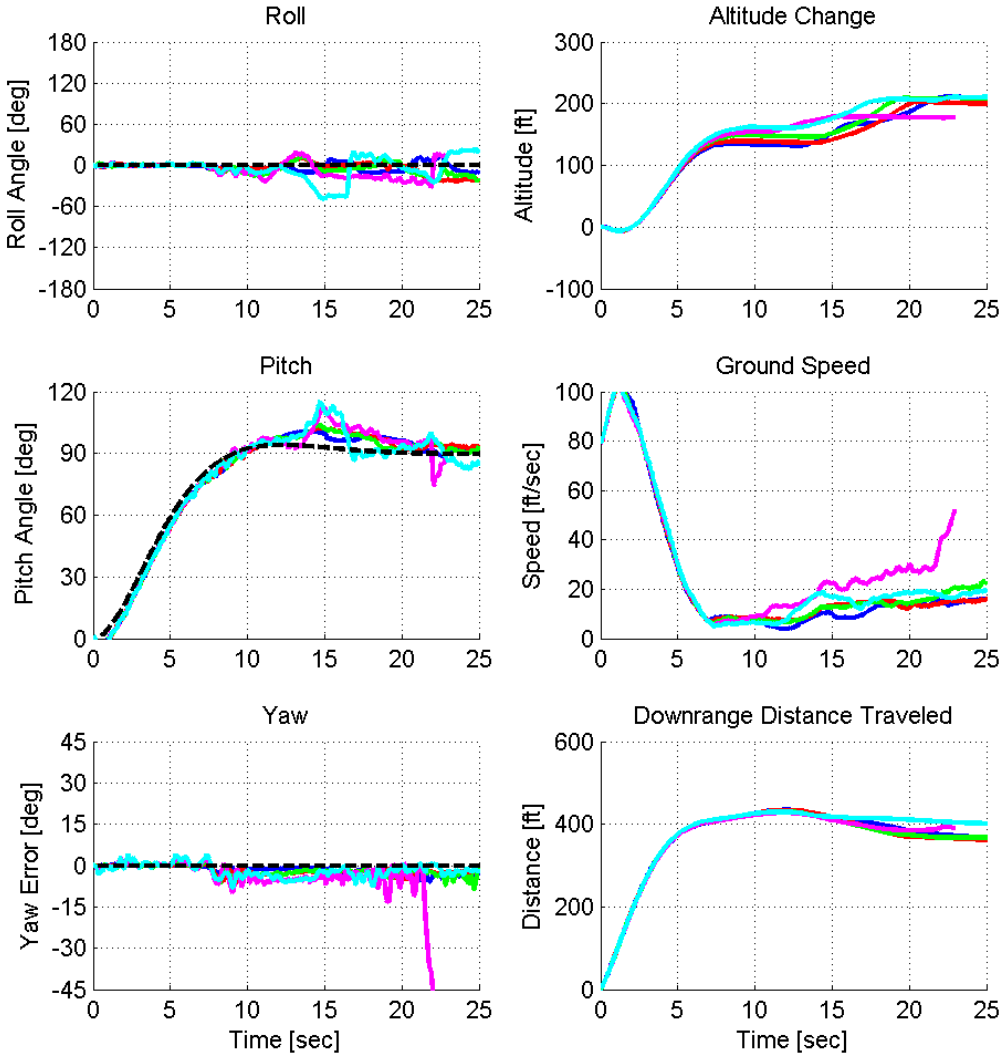
Simulation: MRAC, Rise Time = 5 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	204.7	406.0
2	5.0	yes	200.5	406.3
3	10.0	yes	202.5	402.2
4	15.0	no	210.5	402.7
5	20.0	no	196.7	399.5
	Average		202.6	404.8
	Median		202.5	406.0
	Std. Dev.		2.1	2.3



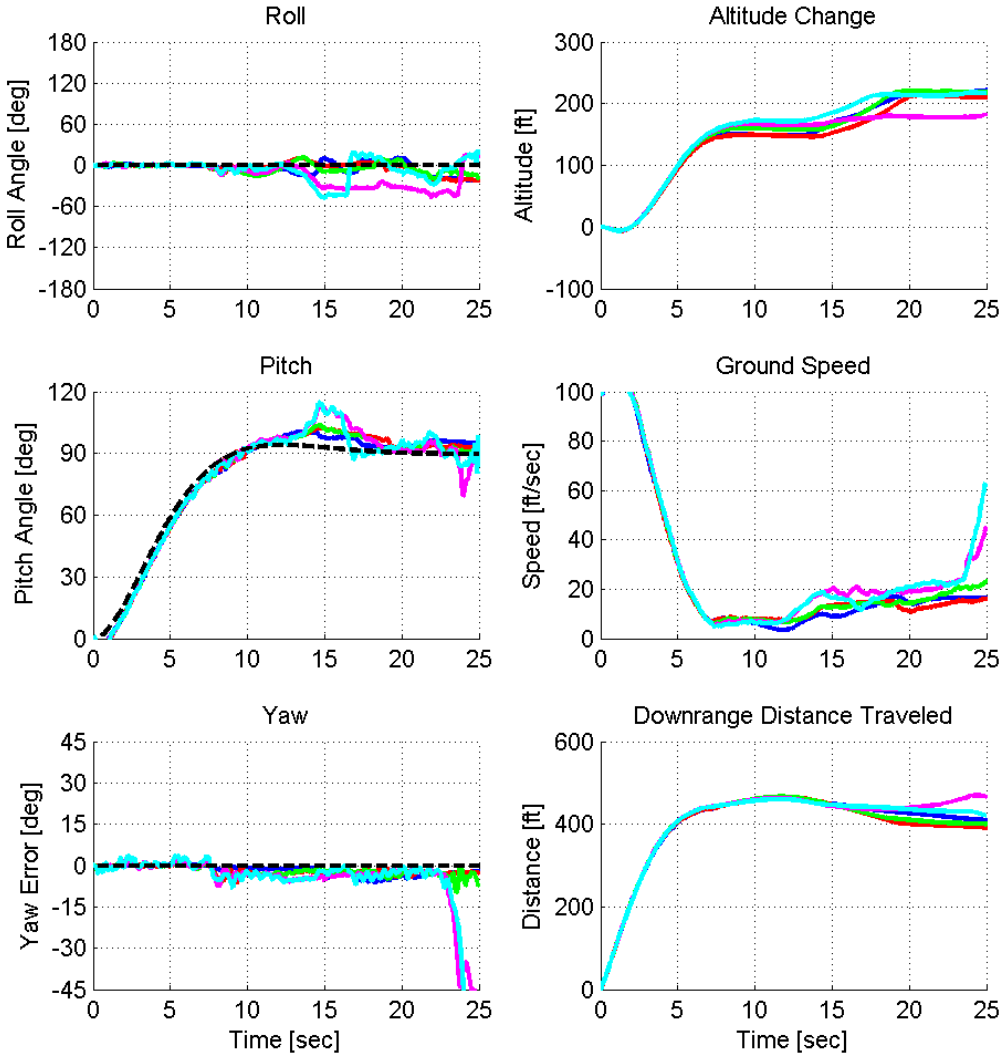
Simulation: MRAC, Rise Time = 5 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	211.1	434.1
2	5.0	yes	202.9	433.7
3	10.0	yes	210.0	430.4
4	15.0	yes	179.0	427.7
5	20.0	no	210.5	429.7
	Average		200.8	431.5
	Median		206.4	432.1
	Std. Dev.		14.9	3.0



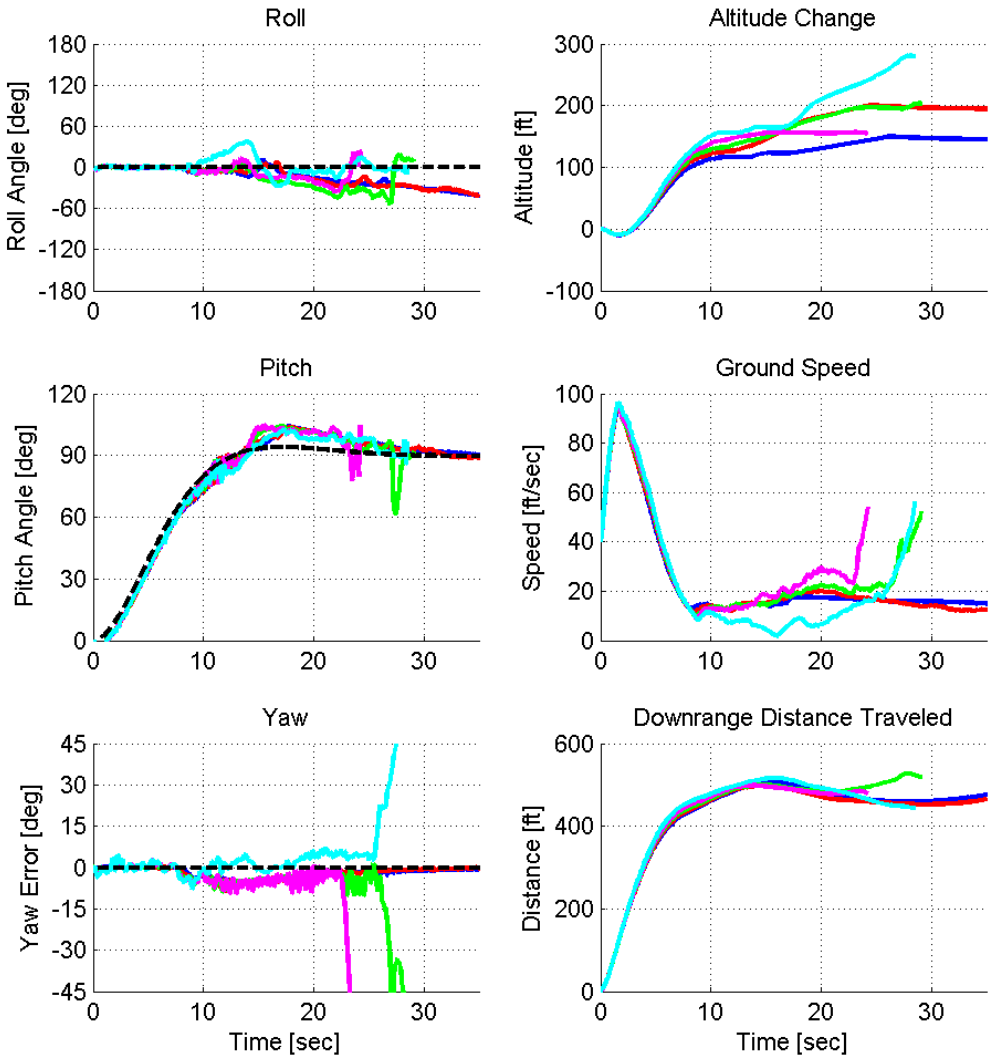
Simulation: MRAC, Rise Time = 5 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	221.5	465.0
2	5.0	yes	213.3	464.9
3	10.0	yes	220.7	466.2
4	15.0	yes	182.2	469.9
5	20.0	yes	217.9	460.2
	Average		211.1	465.3
	Median		217.9	465.0
	Std. Dev.		16.5	3.5



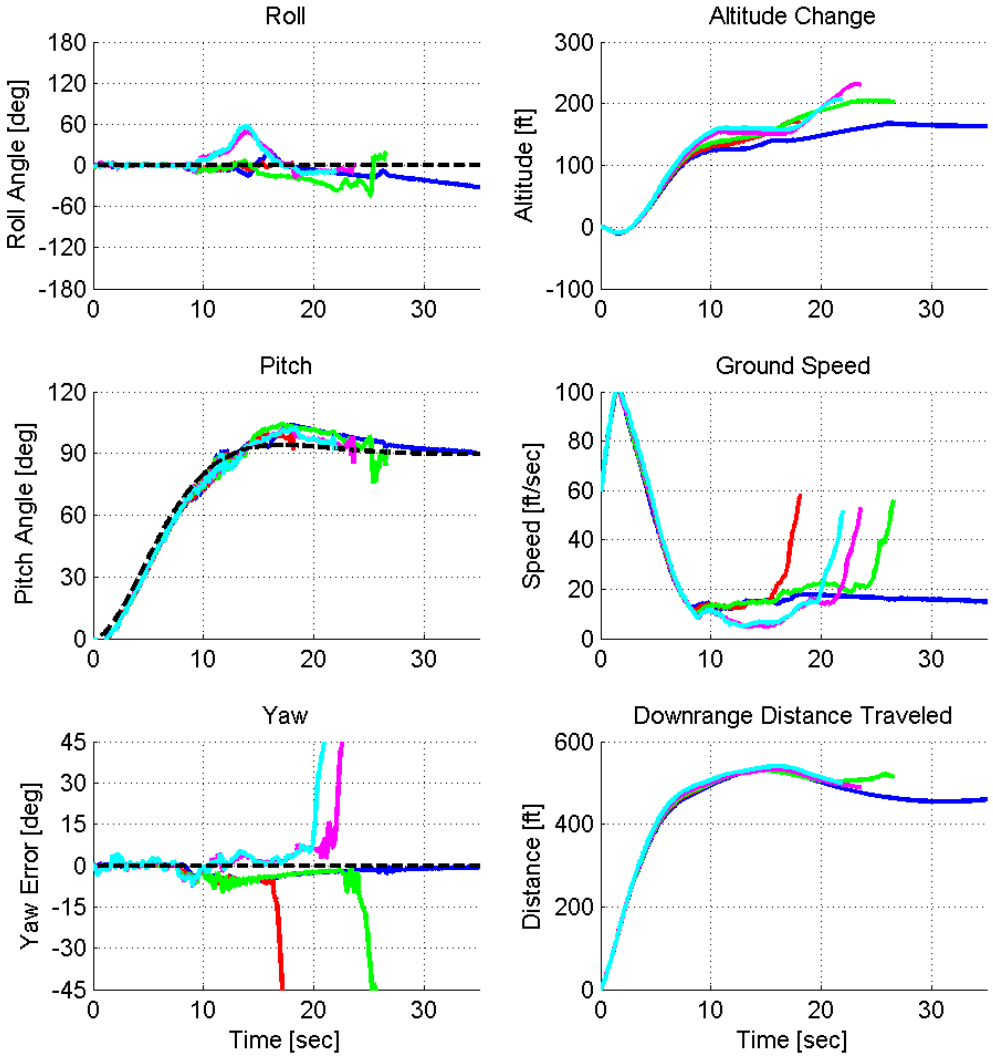
Simulation: MRAC, Rise Time = 7 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	149.8	505.2
2	5.0	yes	200.2	498.0
3	10.0	no	203.8	527.4
4	15.0	no	156.9	498.6
5	20.0	no	280.9	516.7
	Average		175.0	501.6
	Median		175.0	501.6
	Std. Dev.		35.7	5.1



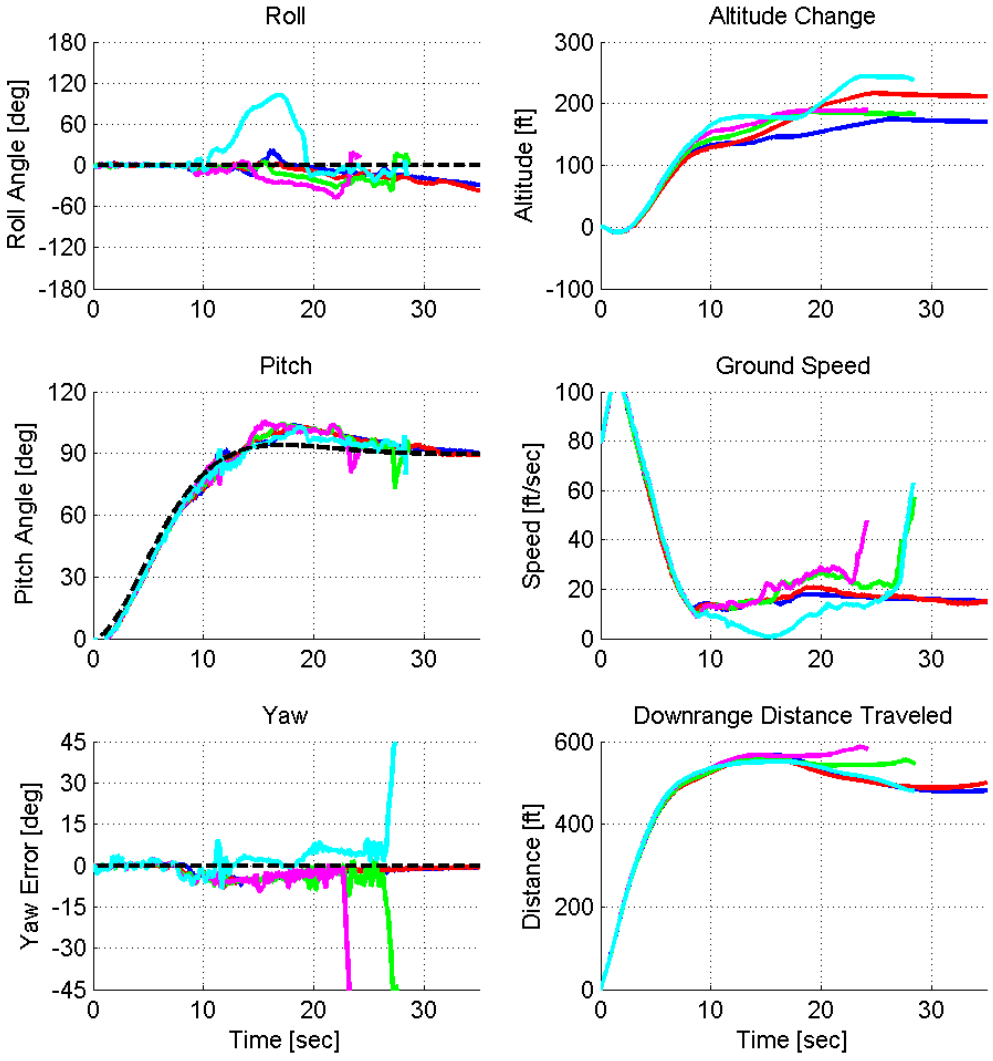
Simulation: MRAC, Rise Time = 7 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	167.6	532.7
2	5.0	no	170.5	532.3
3	10.0	no	204.7	530.3
4	15.0	no	231.6	533.3
5	20.0	no	206.7	540.4
	Average		167.6	532.7
	Median		167.6	532.7
	Std. Dev.		N/A	N/A



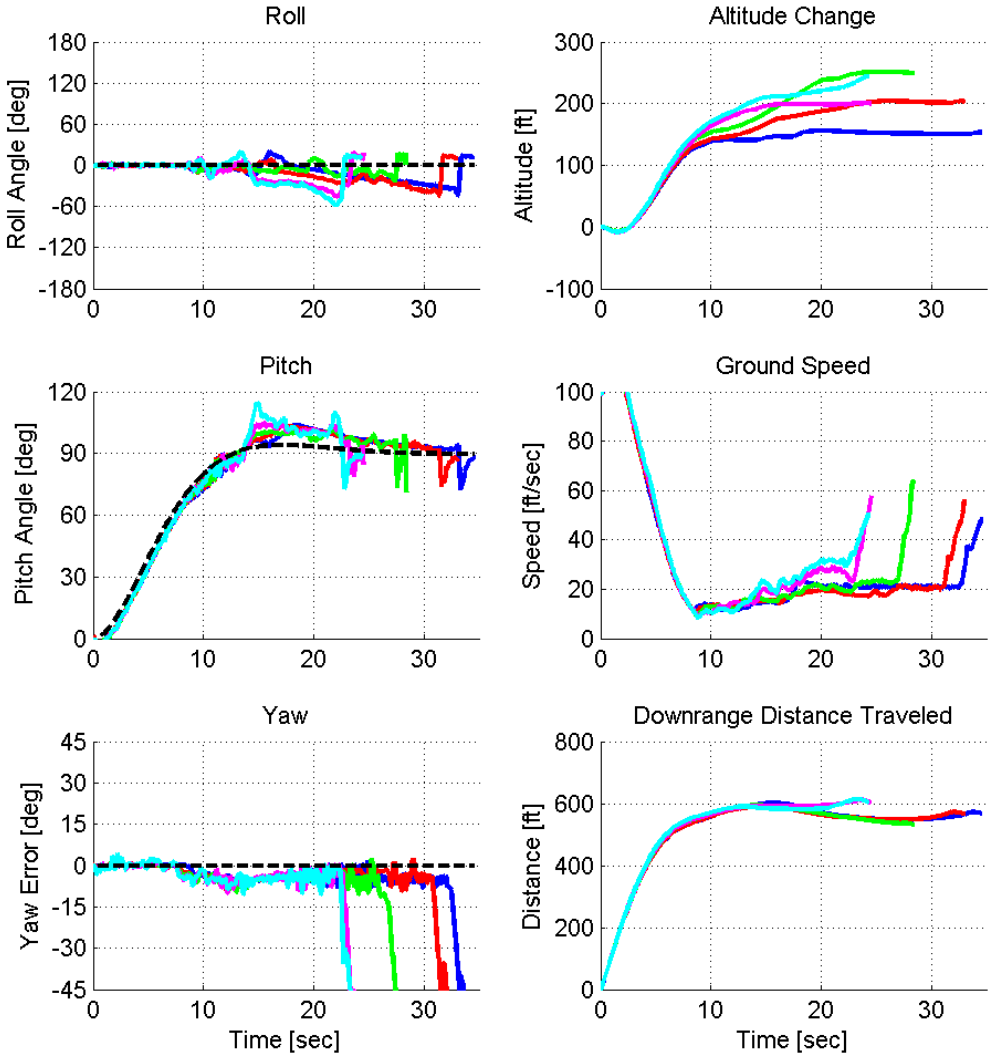
Simulation: MRAC, Rise Time = 7 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	175.1	567.2
2	5.0	yes	216.9	558.8
3	10.0	no	186.9	560.8
4	15.0	no	189.2	586.8
5	20.0	no	244.4	551.1
	Average		196.0	563.0
	Median		196.0	563.0
	Std. Dev.		29.5	5.9



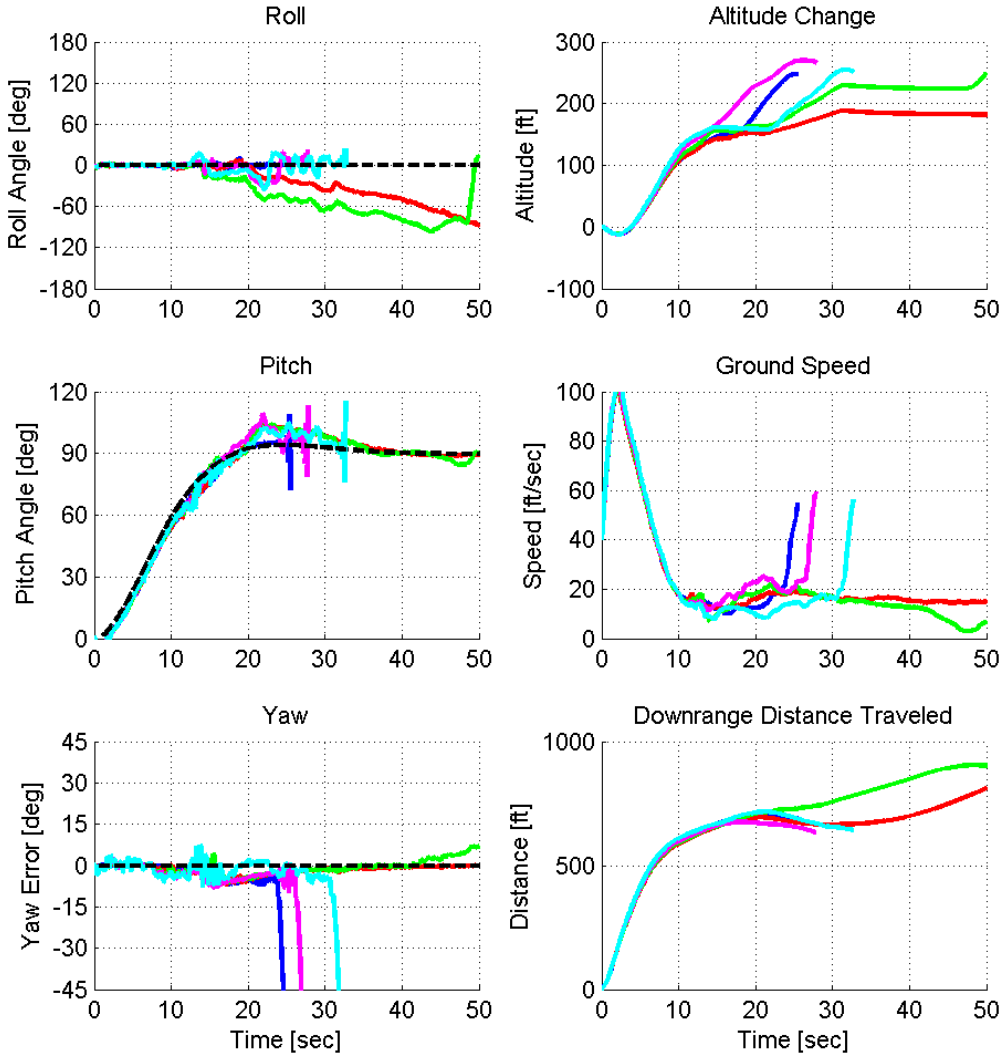
Simulation: MRAC, Rise Time = 7 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	156.2	600.8
2	5.0	no	204.7	591.0
3	10.0	no	251.6	593.4
4	15.0	no	200.8	611.9
5	20.0	no	244.5	614.2
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



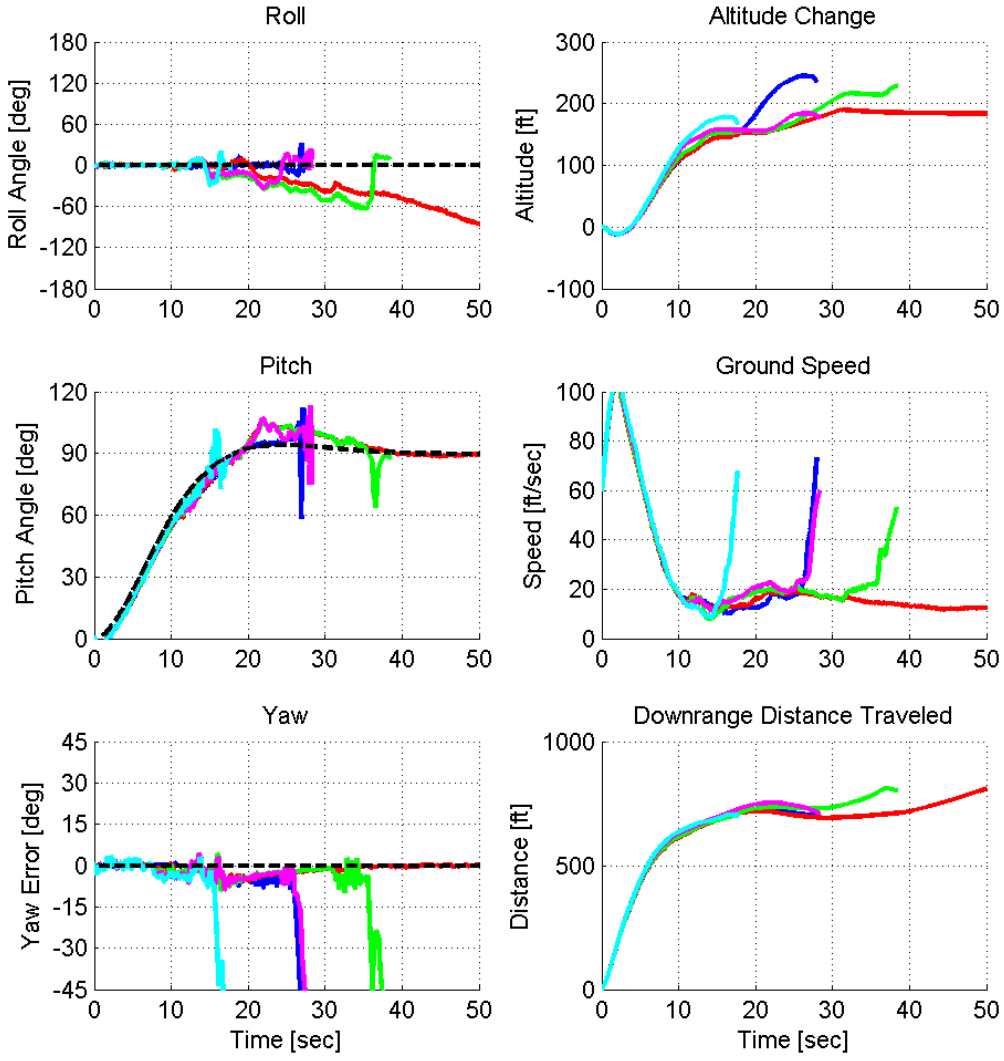
Simulation: MRAC, Rise Time = 10 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	247.7	700.9
2	5.0	yes	188.4	812.4
3	10.0	yes	249.6	904.6
4	15.0	no	269.9	674.8
5	20.0	no	255.0	716.8
	Average		219.0	858.5
	Median		219.0	858.5
	Std. Dev.		43.3	65.2



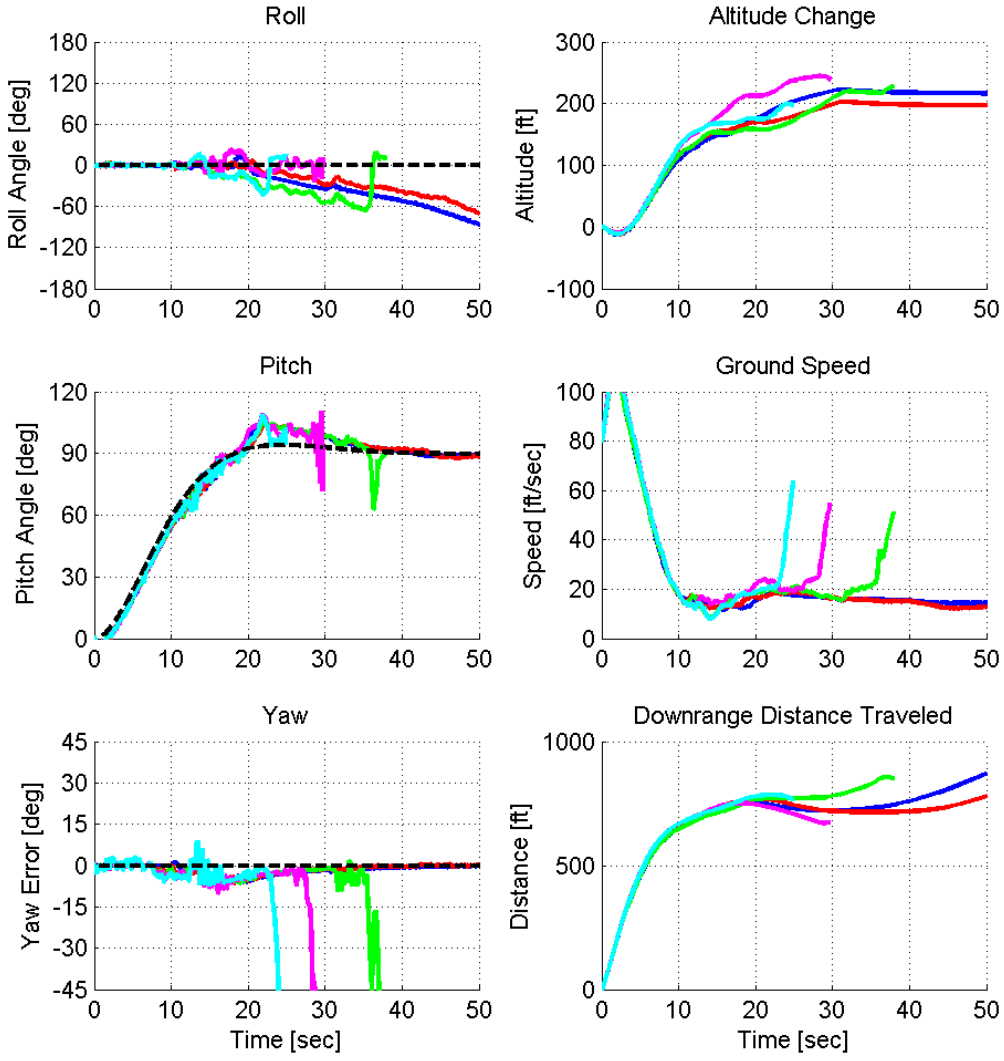
Simulation: MRAC, Rise Time = 10 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	245.0	723.7
2	5.0	yes	189.8	809.9
3	10.0	no	228.3	810.9
4	15.0	no	183.9	753.0
5	20.0	no	178.1	704.0
	Average		189.8	809.9
	Median		189.8	809.9
	Std. Dev.		N/A	N/A



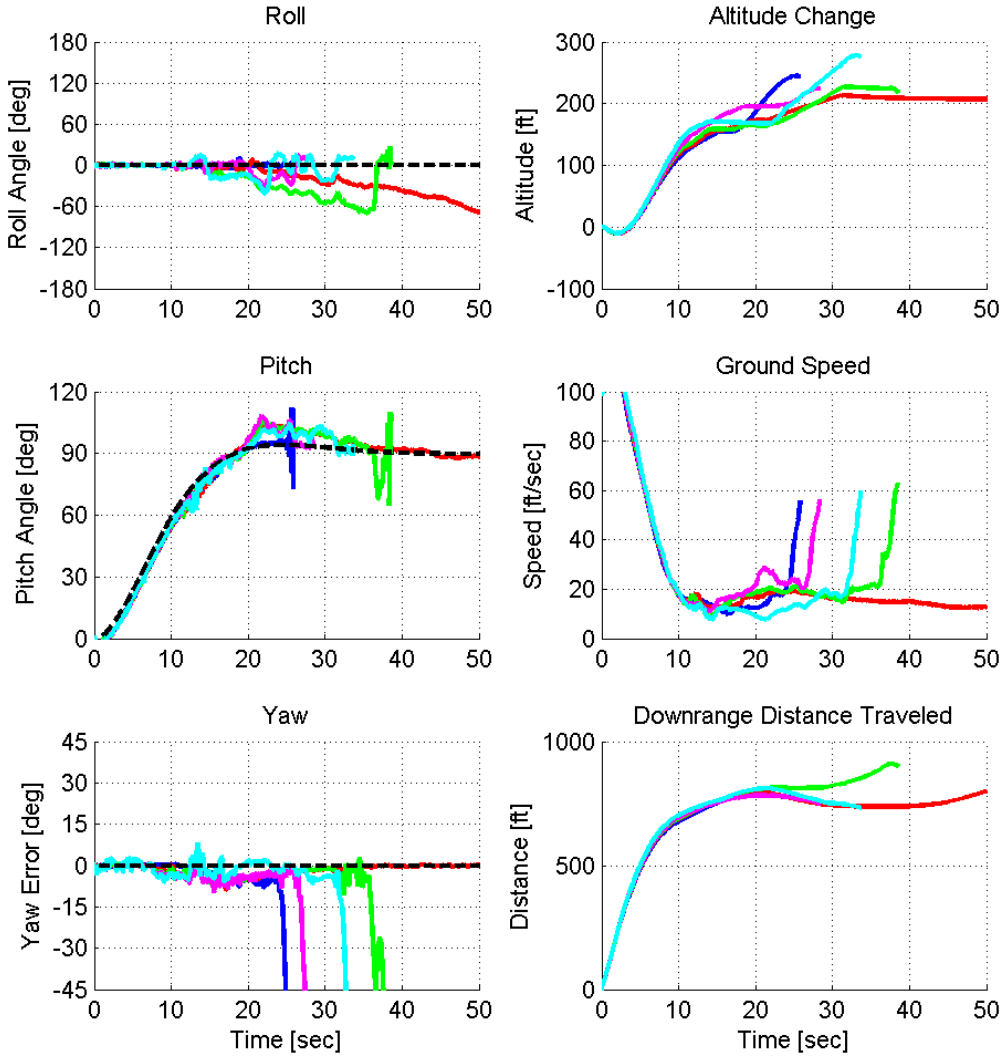
Simulation: MRAC, Rise Time = 10 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	222.5	872.1
2	5.0	yes	203.1	777.8
3	10.0	no	227.8	856.8
4	15.0	no	244.7	751.8
5	20.0	no	199.2	784.2
	Average		212.8	824.9
	Median		212.8	824.9
	Std. Dev.		13.7	66.7



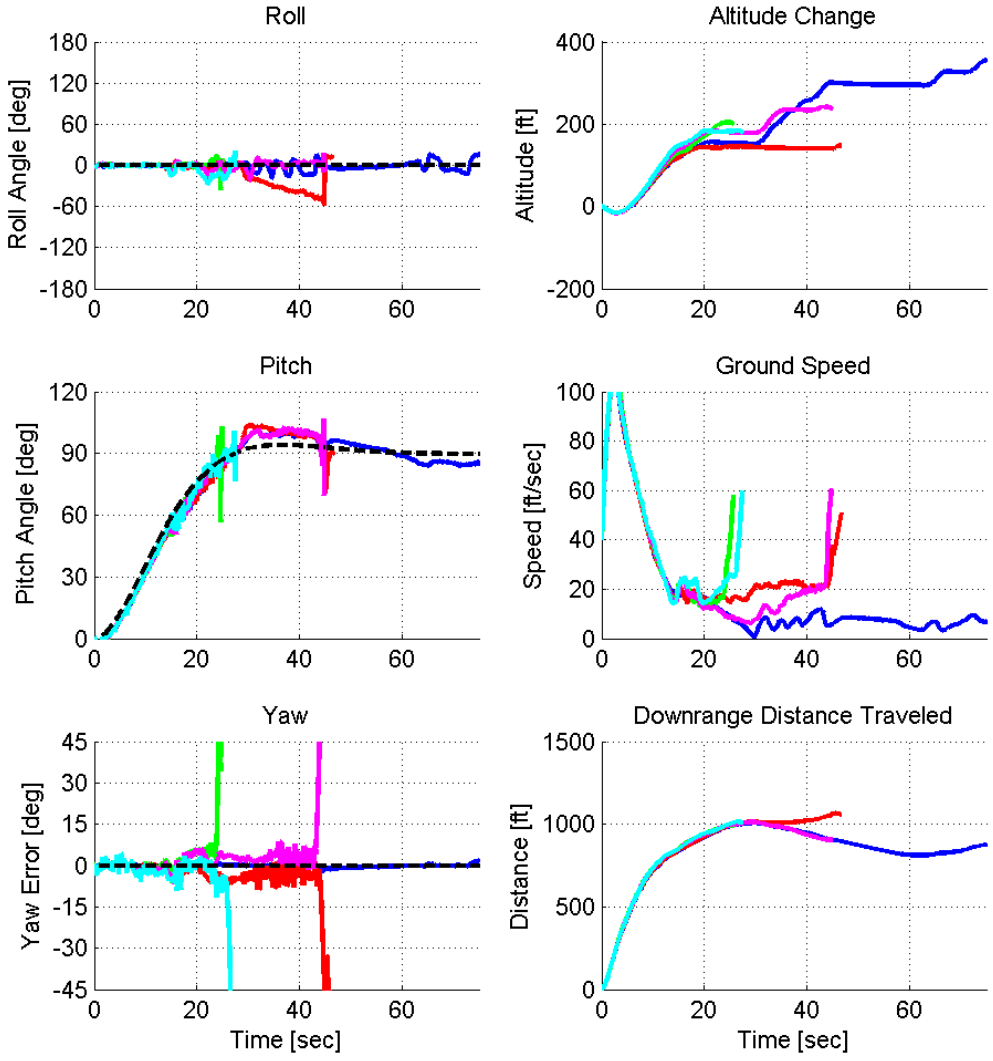
Simulation: MRAC, Rise Time = 10 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	245.3	785.5
2	5.0	yes	213.2	798.3
3	10.0	no	227.0	909.6
4	15.0	no	224.7	782.0
5	20.0	no	277.9	811.0
	Average		213.2	798.3
	Median		213.2	798.3
	Std. Dev.		N/A	N/A



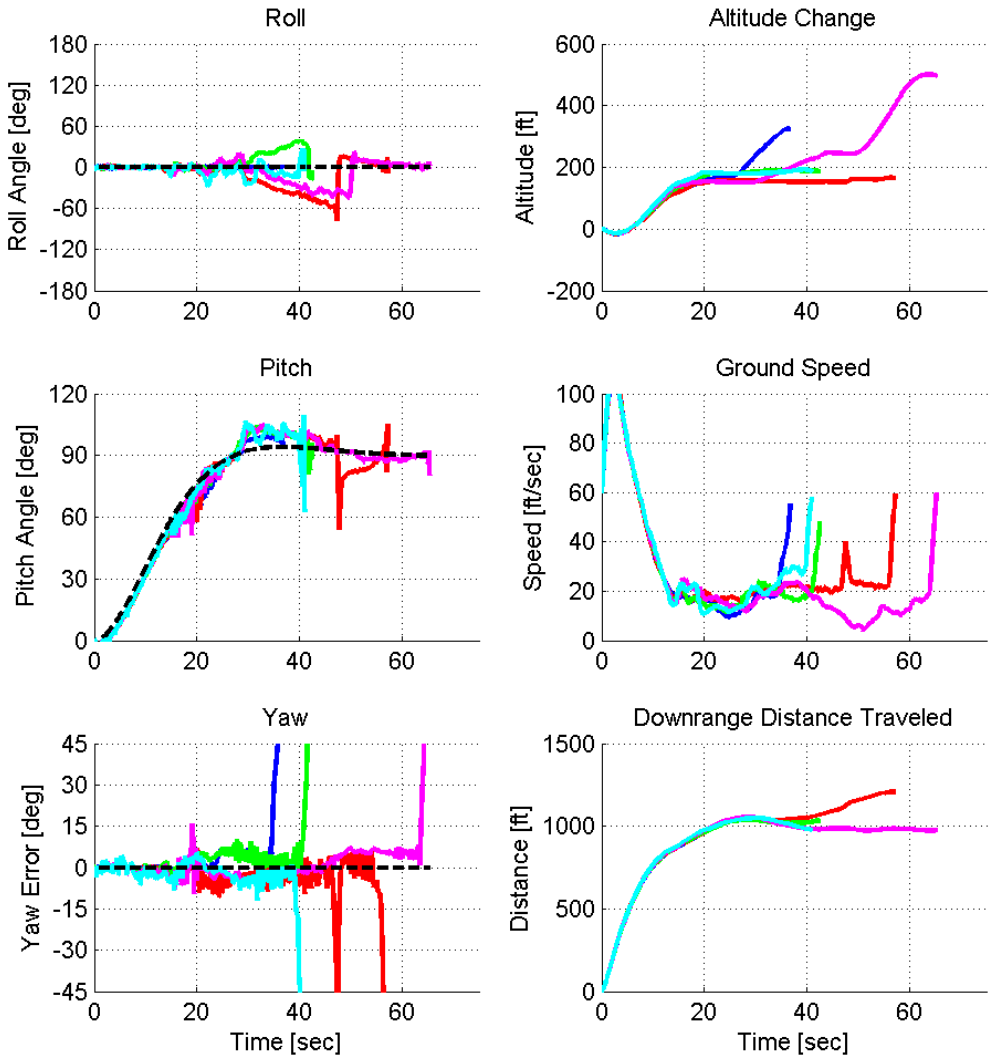
Simulation: MRAC, Rise Time = 15 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	356.2	1008.2
2	5.0	no	148.4	1065.8
3	10.0	no	204.3	993.2
4	15.0	no	242.6	1012.3
5	20.0	no	183.9	1014.3
	Average		356.2	1008.2
	Median		356.2	1008.2
	Std. Dev.		N/A	N/A



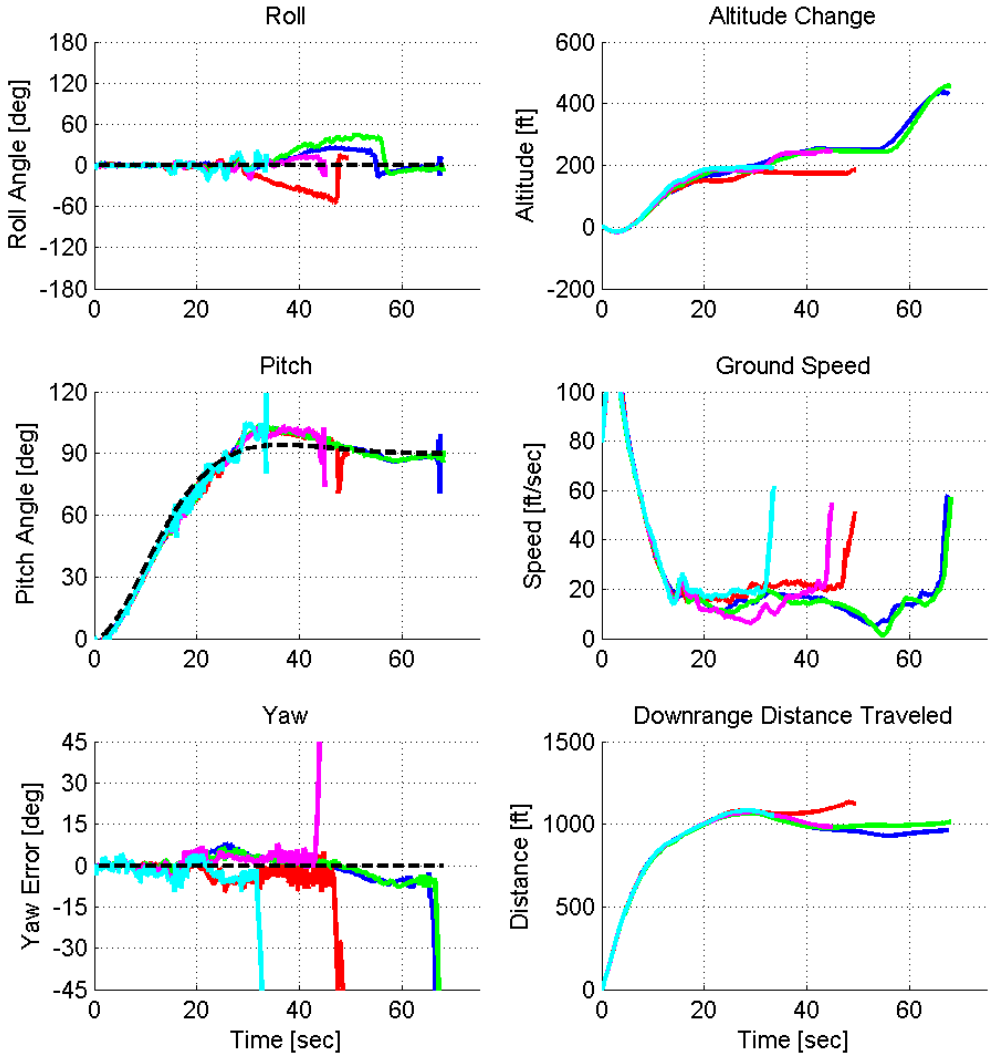
Simulation: MRAC, Rise Time = 15 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	325.5	1042.0
2	5.0	no	166.6	1207.9
3	10.0	no	189.7	1039.7
4	15.0	no	500.6	1057.9
5	20.0	no	192.9	1048.8
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



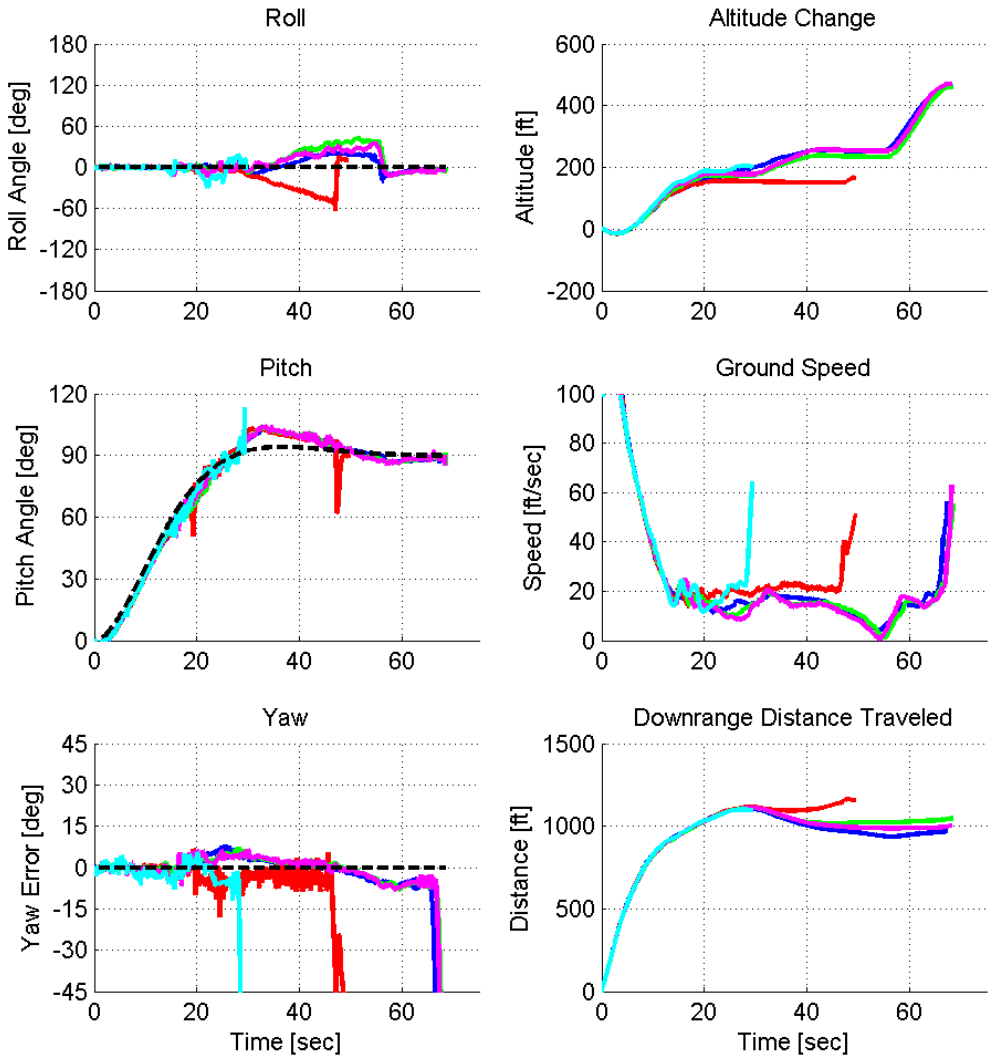
Simulation: MRAC, Rise Time = 15 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	436.5	1076.3
2	5.0	no	184.3	1132.2
3	10.0	no	457.4	1067.7
4	15.0	no	246.4	1075.4
5	20.0	no	193.6	1083.1
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



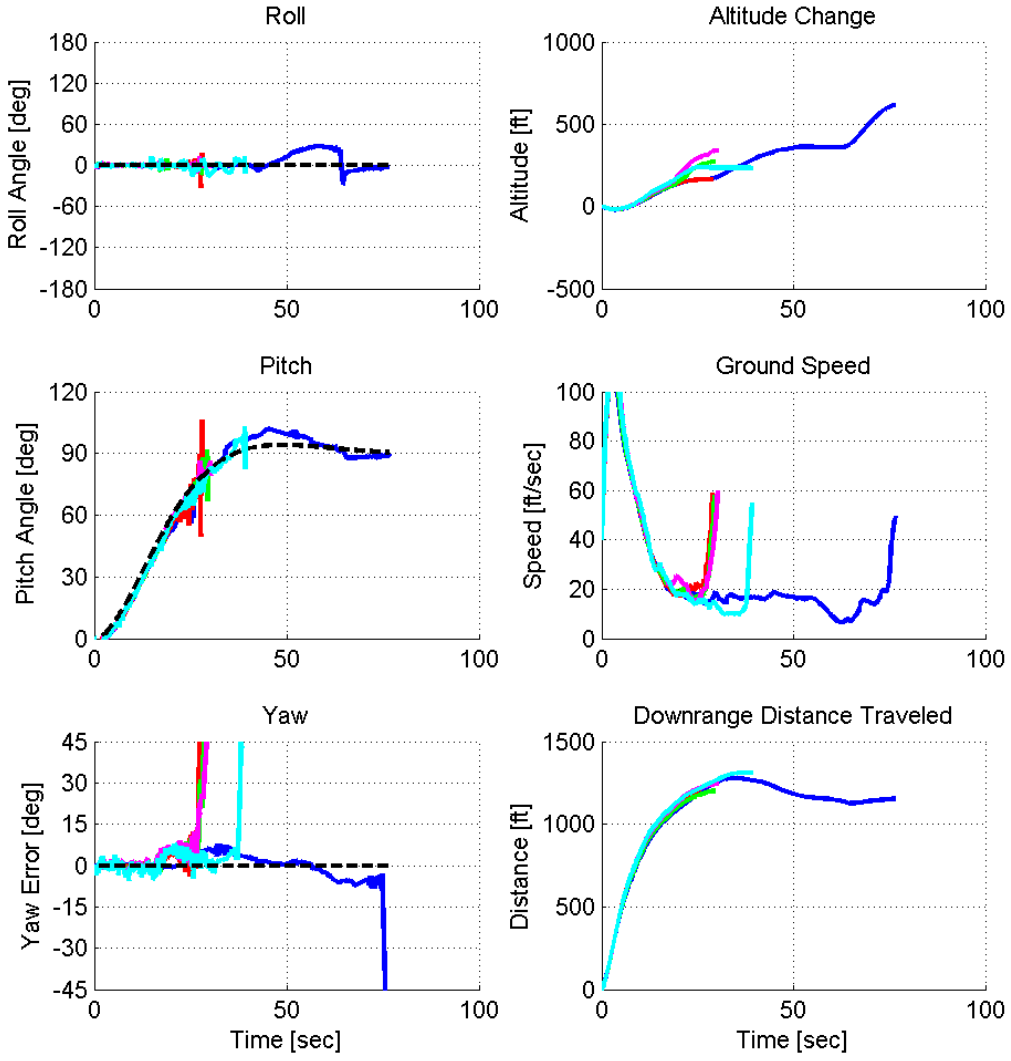
Simulation: MRAC, Rise Time = 15 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	461.5	1106.0
2	5.0	no	167.1	1162.5
3	10.0	no	461.8	1111.5
4	15.0	no	470.2	1111.6
5	20.0	no	204.8	1100.6
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



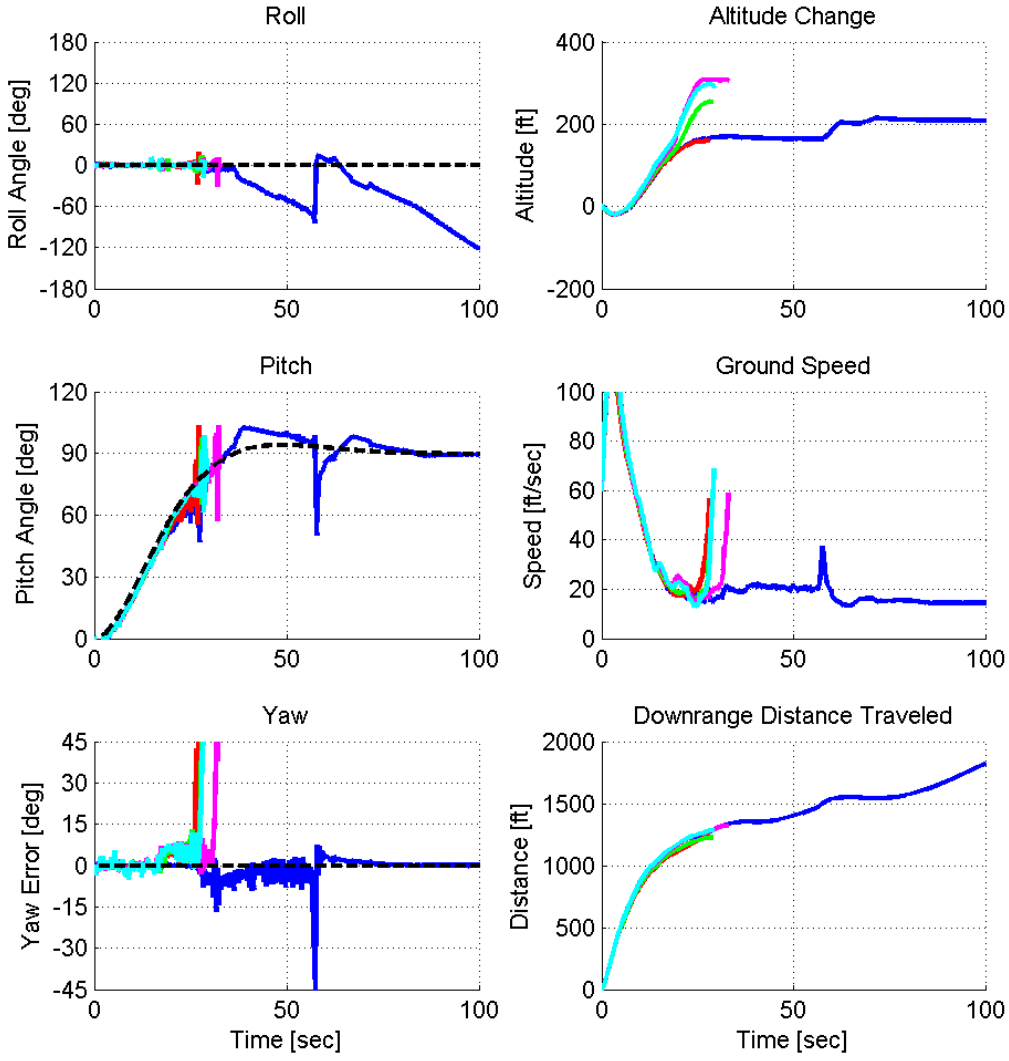
Simulation: MRAC, Rise Time = 20 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	616.2	1278.5
2	5.0	no	164.6	1239.3
3	10.0	no	272.8	1201.7
4	15.0	no	340.7	1245.3
5	20.0	no	238.4	1311.9
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



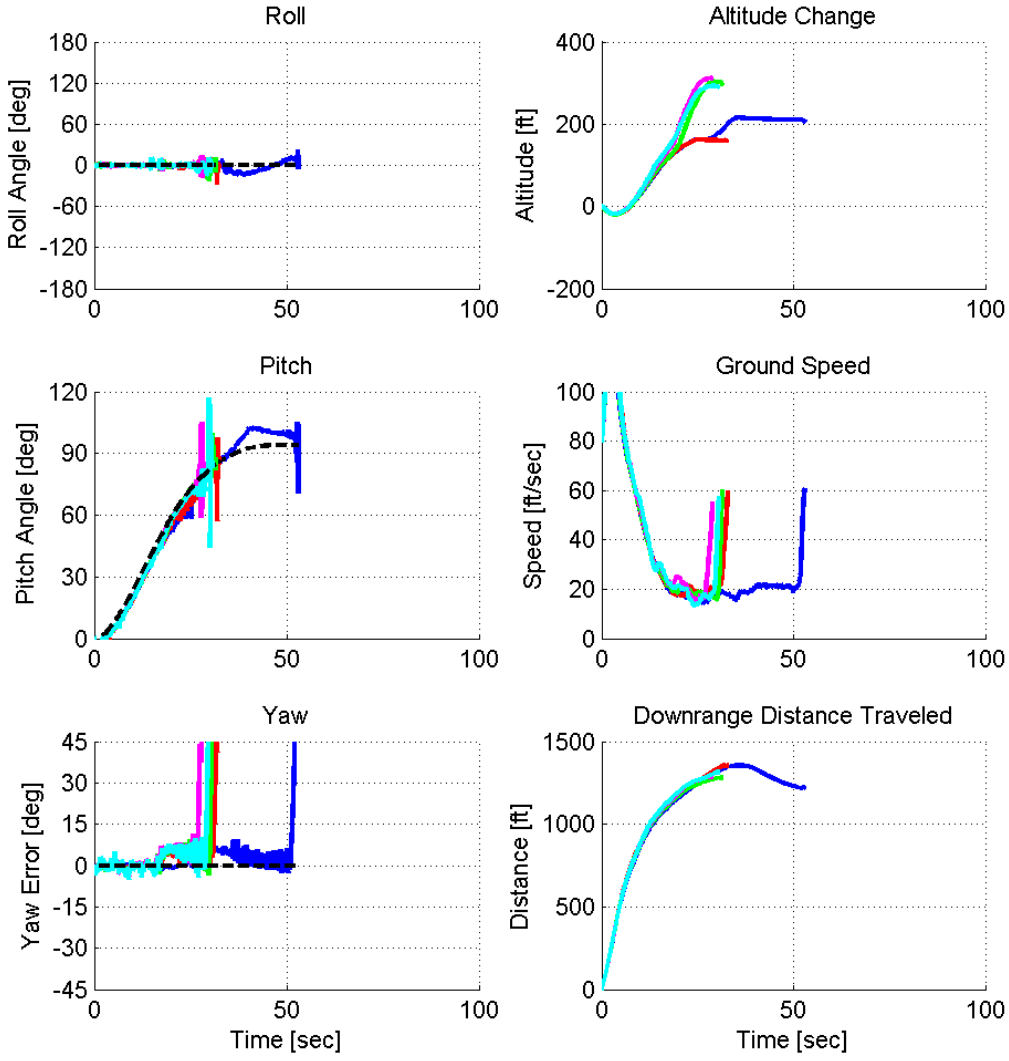
Simulation: MRAC, Rise Time = 20 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	215.1	1822.9
2	5.0	no	160.0	1233.2
3	10.0	no	255.3	1225.5
4	15.0	no	308.4	1321.6
5	20.0	no	297.2	1285.1
	Average		215.1	1822.9
	Median		215.1	1822.9
	Std. Dev.		N/A	N/A



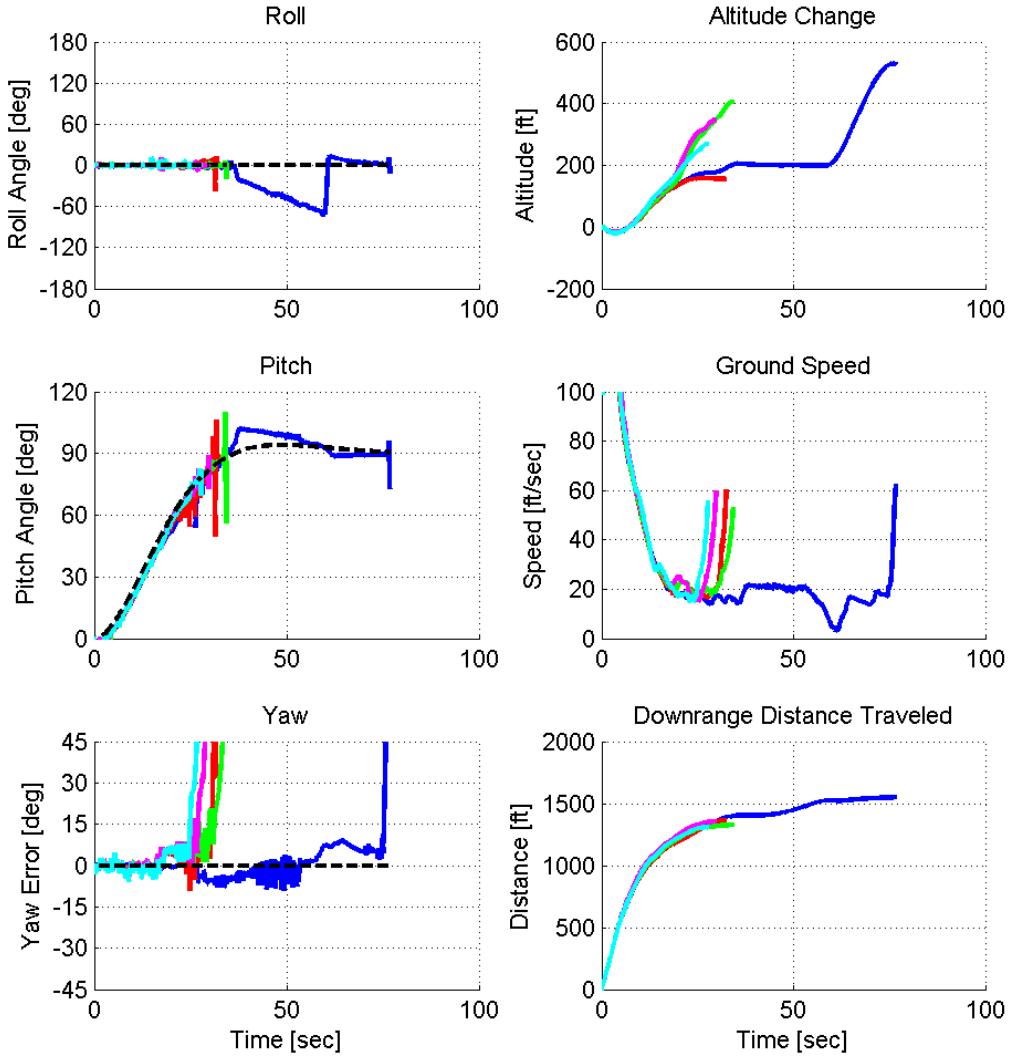
Simulation: MRAC, Rise Time = 20 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	215.9	1354.8
2	5.0	no	163.1	1353.9
3	10.0	no	303.5	1281.2
4	15.0	no	312.2	1287.7
5	20.0	no	292.5	1315.0
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



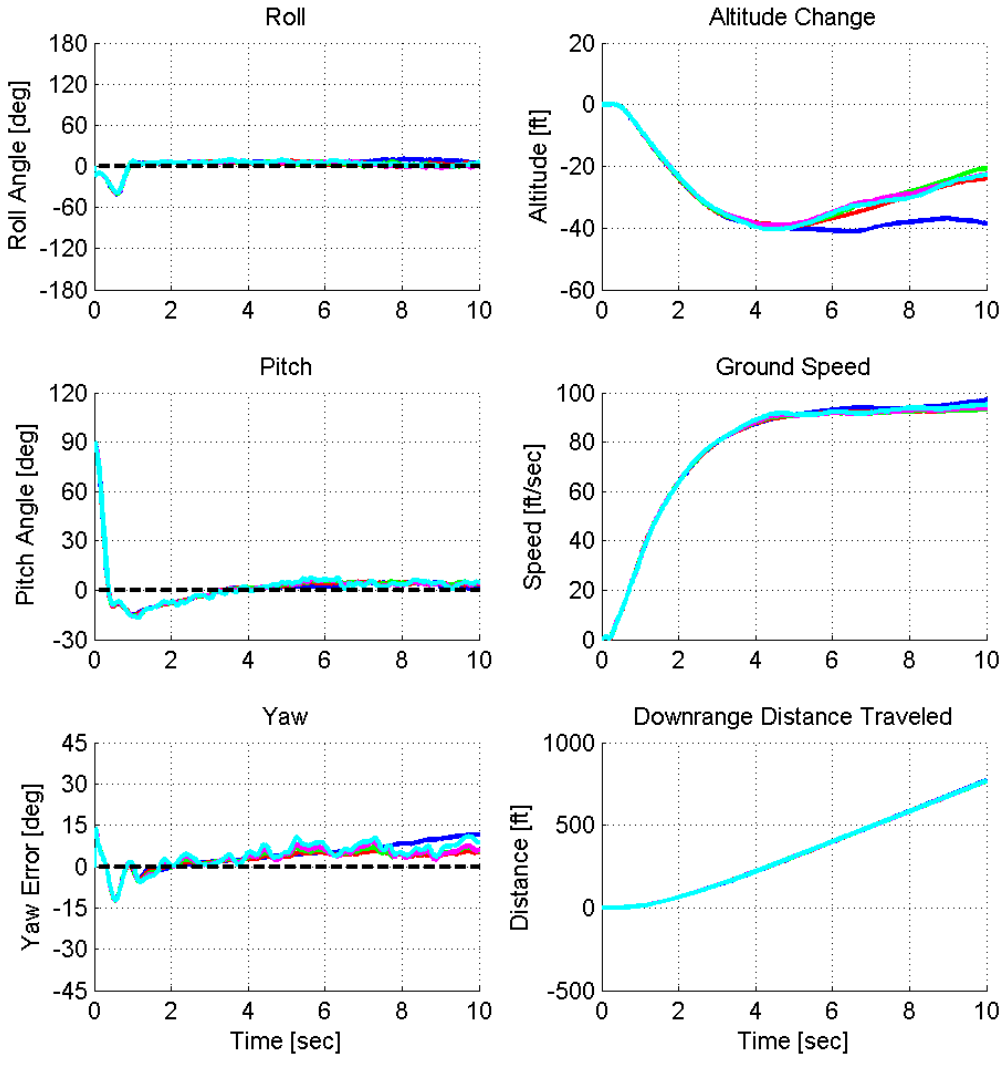
Simulation: MRAC, Rise Time = 20 sec

Approach Speed: 100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	530.0	1550.7
2	5.0	no	158.7	1365.3
3	10.0	no	405.6	1327.2
4	15.0	no	347.4	1357.0
5	20.0	no	267.8	1310.4
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



A.4 Hover-to-Level Transition

Hover-to-Level Flight			
Test No.	Wind [ft/sec]	Successful?	Max. Altitude Change [ft]
1	0.0	yes	-41.1
2	5.0	yes	-39.2
3	10.0	yes	-39.3
4	15.0	yes	-39.1
5	20.0	yes	-40.4
	Average		-39.8
	Median		-39.3
	Std. Dev.		0.9



B FLIGHT TEST RESULTS

Complete results for the 340 separate transition-to-hover flight tests that were run as part of this thesis are presented here. Plots showing the time history of the aircraft's attitude, altitude, ground speed, and downrange distance traveled are included along with data for each flight such as the wind speed, maximum altitude change, maximum distance traveled, and whether or not the airplane was able to achieve and maintain stable hover throughout the duration of the flight test.

Each of the three hover flight modes were tested under different conditions to see how well they handled different types of transitions. For the HOVER_PID_REF and HOVER_ADAPTIVE flight modes, each combination of controller type and reference model design was tested at four different approach speeds: 40, 60, 80, ∼100 ft/sec. Each controller, reference model, and approach speed combination was then repeated 5 times in order to get a measure of the repeatability of the results. The 100 ft/sec tests are labeled as an approximate approach speed because they are run at maximum throttle. Since the recorded initial speed is actually a GPS ground speed and not true airspeed, a strong headwind could sometimes prevent the aircraft from ever being able to reach the desired 100 ft/sec, in which case the transition was initiated once the aircraft had reached a maximum stead state velocity. Additionally, throttle controller input was always set to 0 ft/sec descent/climb rate for the entire duration of the tests so variations in altitude are caused by either the bleeding off of excess kinetic energy or from the hover divergence logic controller needing to increase throttle in order to maintain stable hover, and not the pilot manually changing the desired descent/climb rate.

The plots for the aircraft's pitch and roll angles are in reference to the Earth

frame (i.e. 0 roll and 0 pitch mean the nose and wings are level to the horizon) whereas yaw is plotted as an error value referenced to the initial heading so that the different tests can be compared more easily (i.e. 0 yaw error means the aircraft has the same heading as when the transition maneuver was initiated). Altitude and distance traveled are also with reference to the position of the airplane at the start of the test instead of to some predetermined reference coordinate, so that every plot starts at zero altitude and zero distance traveled. It is important to note the ground speed is measured without any reference to the direction the aircraft is moving. It is simply the absolute distance traveled since the last GPS update divided by the time since the last update, so the value is always positive. The downrange distance traveled is also measured as the projection of the aircraft's actual path onto the vector that originates at the airplane's initial position and continues outward in the direction of the airplane's initial heading. Measuring the distance traveled in this way eliminates any lateral drifting due to wind from the measurement, which can occur even if the the aircraft is able to maintain zero yaw error throughout the test.

The colors that represent the plots for each of the 5 repeated tests per flight condition combination are as follows:

Flight No.	Plot Color	Wind Speed [ft/sec]
Ref. Model	Dashed Black	N/A
1	Solid Blue	Measured
2	Solid Red	Measured
3	Solid Green	Measured
4	Solid Magenta	Measured
5	Solid Cyan	Measured

Unlike the simulations, where the wind speed could be specified, the wind speed during the flight tests was dependent on the day and simply had to be measured using a hand-held anemometer. The recorded wind speed for each test was the peak wind speed measured by the anemometer during the duration of the individual test.

Finally, a summary table is presented at the beginning of each controller type sec-

tion which contains the probability of successful transition and the average, median, and standard deviation values for both the maximum altitude change, and maximum distance traveled for each combination of controller, reference model and approach speed.

B.1 PID Control Step Response

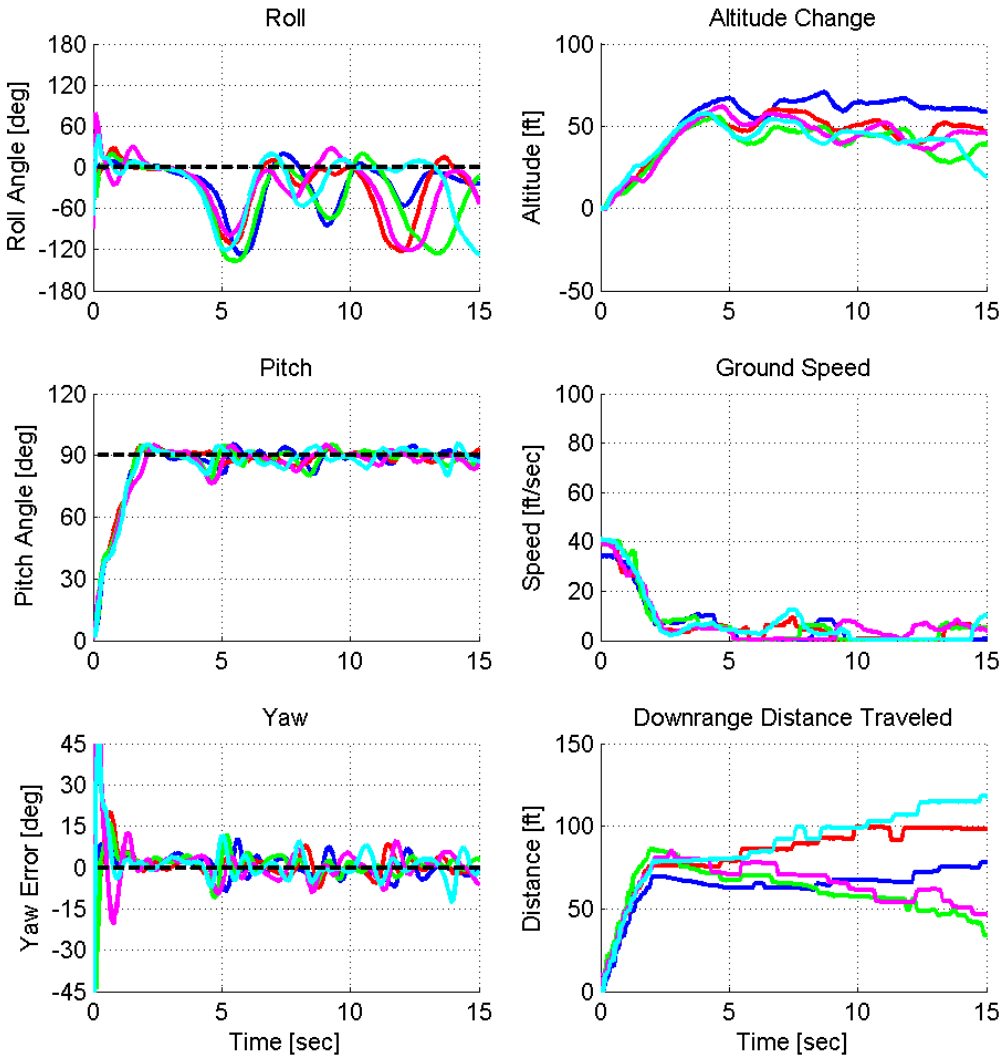
PID Control Step Response Flight Tests: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
Step	100%	100%	100%	100%	100%
Cumulative	100%	100%	100%	100%	100%

PID Control Step Response Flight Tests			
Approach Speed	Max. Altitude Change [ft]		
	Average	Median	Std. Dev.
40 ft/sec	61.2	60.3	5.7
60 ft/sec	83.4	84.7	14.9
80 ft/sec	211.8	205.7	36.3
100 ft/sec	240.5	201.9	131.8

PID Control Step Response Flight Tests			
Approach Speed	Max. Distance Traveled [ft]		
	Average	Median	Std. Dev.
40 ft/sec	93.2	85.9	16.0
60 ft/sec	99.5	103.5	14.1
80 ft/sec	133.2	131.1	10.4
100 ft/sec	114.6	122.2	13.8

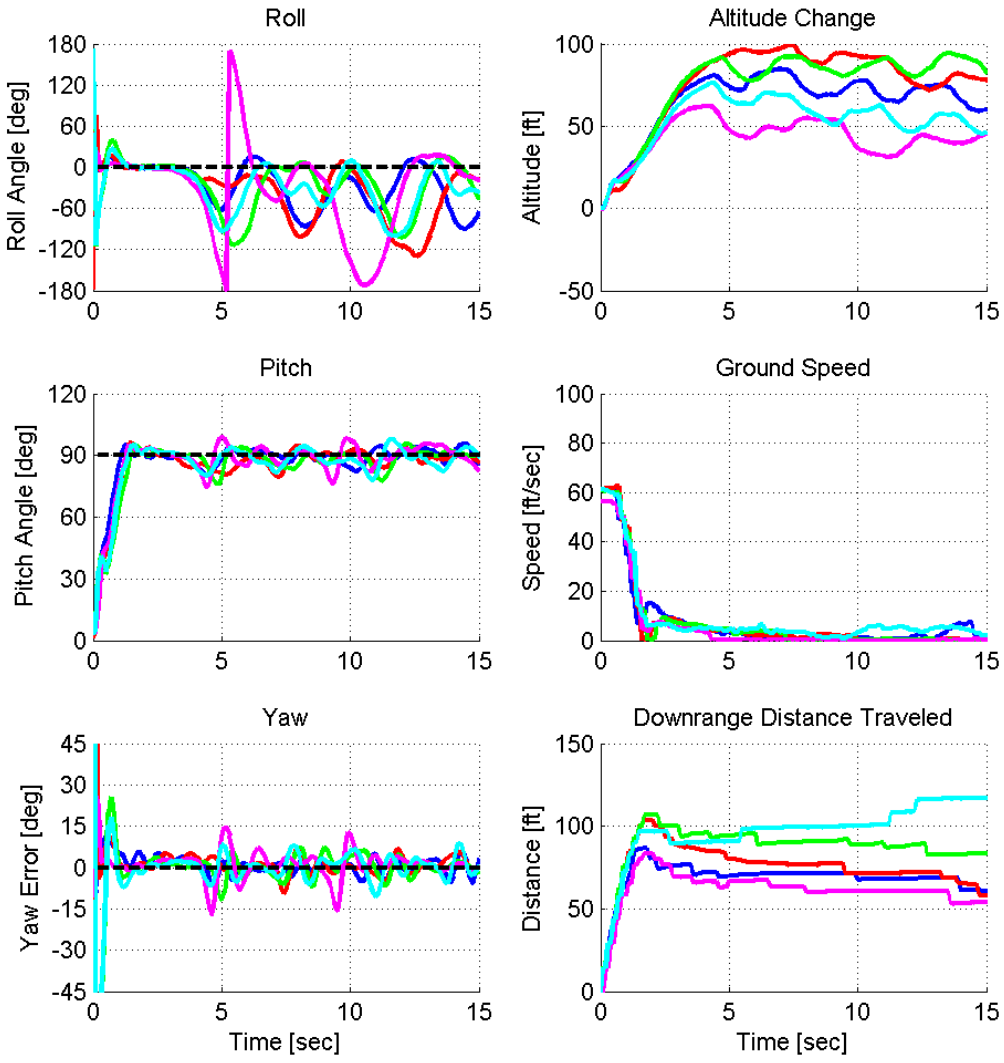
Flight Test: PID Step Response

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	2.6	yes	70.4	78.1
2	0.0	yes	60.3	99.5
3	3.6	yes	55.4	85.9
4	1.3	yes	61.9	84.4
5	0.0	yes	57.7	118.2
	Average		61.2	93.2
	Median		60.3	85.9
	Std. Dev.		5.7	16.0



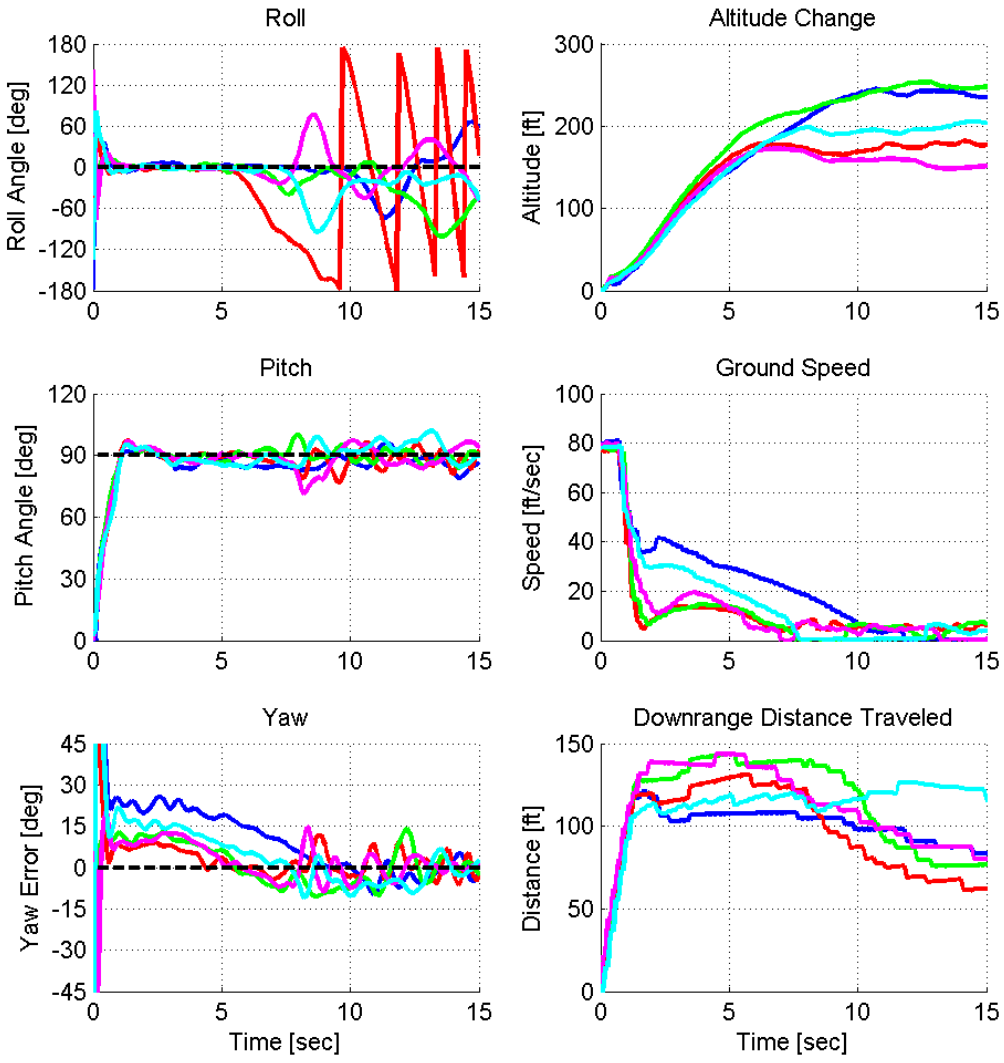
Flight Test: PID Step Response

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	84.7	86.6
2	2.0	yes	99.5	103.5
3	3.3	yes	94.5	106.7
4	4.3	yes	62.1	83.7
5	0.0	yes	76.2	117.2
	Average		83.4	99.5
	Median		84.7	103.5
	Std. Dev.		14.9	14.1



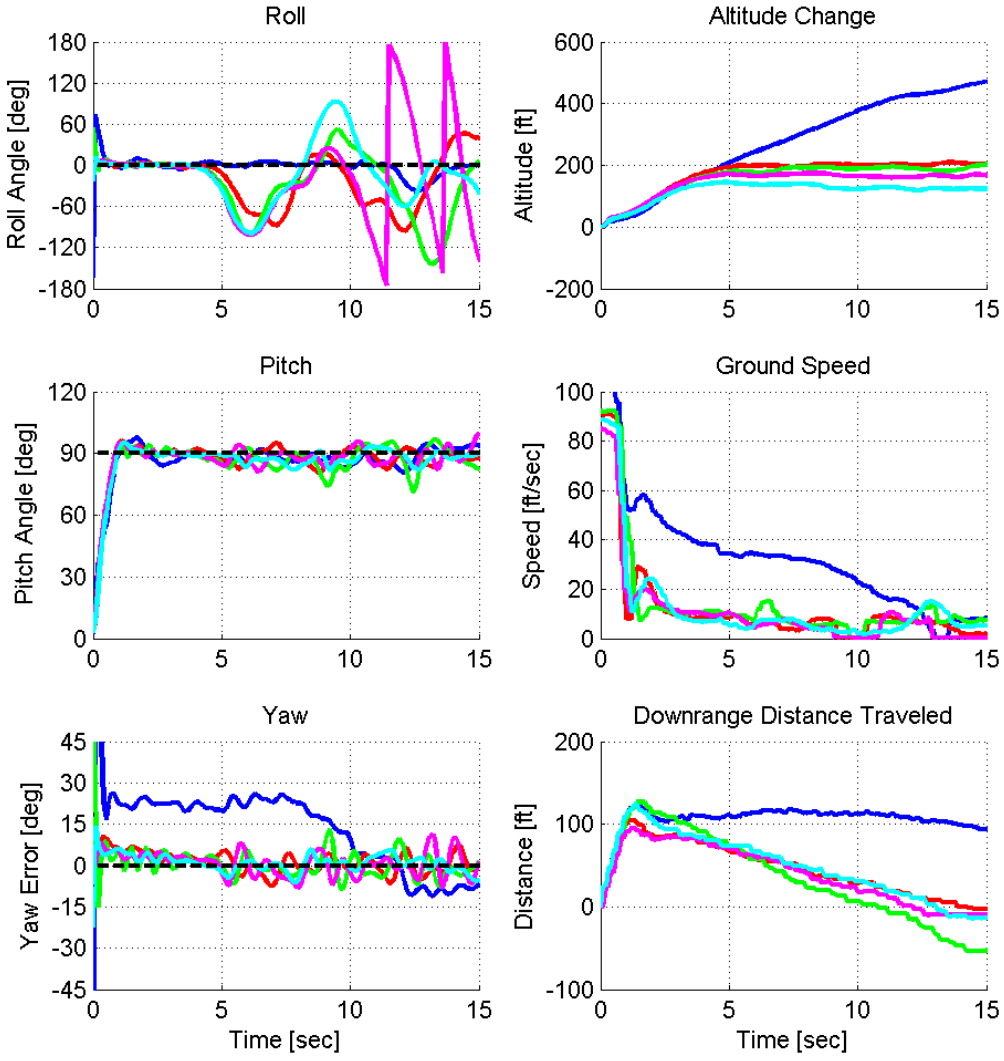
Flight Test: PID Step Response

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	3.9	yes	244.9	120.7
2	6.2	yes	182.2	131.1
3	2.3	yes	253.3	143.9
4	0.0	yes	172.7	143.8
5	7.5	yes	205.7	126.4
	Average		211.8	133.2
	Median		205.7	131.1
	Std. Dev.		36.3	10.4



Flight Test: PID Step Response

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	11.2	yes	471.8	122.2
2	4.9	yes	209.9	104.5
3	7.2	yes	201.9	126.9
4	4.6	yes	174.4	95.4
5	0.0	yes	144.6	123.7
	Average		240.5	114.6
	Median		201.9	122.2
	Std. Dev.		131.8	13.8



B.2 PID Control Reference Model Response

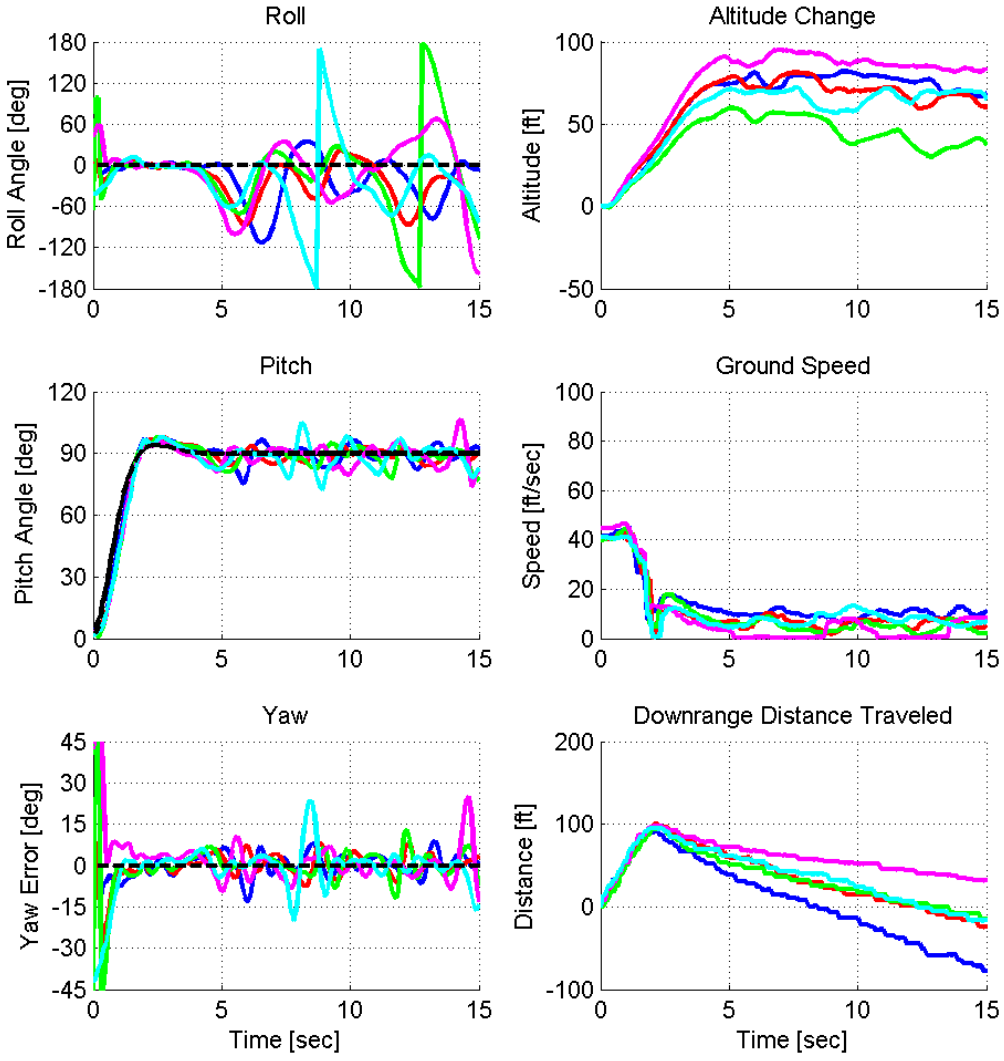
PID Control Reference Model Response Flight Tests: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	80%	100%	100%	100%	95%
5	60%	80%	80%	100%	80%
7	100%	80%	100%	40%	80%
10	100%	100%	80%	80%	90%
15	100%	100%	100%	100%	100%
20	100%	100%	100%	80%	95%
Cumulative	93%	95%	95%	88%	92.5%

PID Control Reference Model Response Flight Tests				
Rise Time	Approach Speed	Max. Altitude Change [ft]		
		Average	Median	Std. Dev.
1 sec	40 ft/sec	78.1	81.4	13.1
	60 ft/sec	183.1	182.0	29.2
	80 ft/sec	182.9	161.4	60.6
	100 ft/sec	139.9	126.0	27.6
2 sec	40 ft/sec	144.5	117.8	68.3
	60 ft/sec	145.9	151.1	15.6
	80 ft/sec	193.8	184.5	32.1
	100 ft/sec	194.6	181.7	19.1
3 sec	40 ft/sec	119.8	138.0	50.2
	60 ft/sec	124.8	102.5	53.6
	80 ft/sec	142.8	127.7	59.8
	100 ft/sec	195.0	186.9	41.7
5 sec	40 ft/sec	96.8	101.4	23.7
	60 ft/sec	102.2	105.1	11.5
	80 ft/sec	132.3	130.5	9.0
	100 ft/sec	216.0	190.2	91.1
7 sec	40 ft/sec	133.6	127.3	31.2
	60 ft/sec	207.9	211.0	50.0
	80 ft/sec	150.8	140.3	29.4
	100 ft/sec	135.8	135.8	30.5
10 sec	40 ft/sec	207.3	187.5	82.9
	60 ft/sec	207.3	173.5	111.4
	80 ft/sec	202.9	196.9	21.5
	100 ft/sec	276.3	254.2	61.4
15 sec	40 ft/sec	402.0	278.1	200.9
	60 ft/sec	410.6	419.3	127.9
	80 ft/sec	436.8	469.4	215.4
	100 ft/sec	498.3	498.9	138.5
20 sec	40 ft/sec	434.9	363.9	212.4
	60 ft/sec	482.0	483.8	46.8
	80 ft/sec	497.4	469.9	217.1
	100 ft/sec	684.4	542.3	405.8

PID Control Reference Model Response Flight Tests				
Rise Time	Approach Speed	Max. Distance Traveled [ft]		
		Average	Median	Std. Dev.
1 sec	40 ft/sec	95.5	94.6	3.5
	60 ft/sec	155.4	153.4	13.5
	80 ft/sec	183.4	179.5	31.7
	100 ft/sec	172.6	185.8	23.4
2 sec	40 ft/sec	113.2	115.3	4.5
	60 ft/sec	144.9	149.3	11.9
	80 ft/sec	149.5	156.3	12.4
	100 ft/sec	122.5	111.4	30.4
3 sec	40 ft/sec	162.0	162.2	14.7
	60 ft/sec	276.3	282.2	30.8
	80 ft/sec	295.5	295.2	22.1
	100 ft/sec	347.7	383.8	574
5 sec	40 ft/sec	281.6	245.0	67.8
	60 ft/sec	344.2	330.7	56.7
	80 ft/sec	415.4	410.3	37.5
	100 ft/sec	434.8	382.8	146.8
7 sec	40 ft/sec	337.6	316.5	65.3
	60 ft/sec	393.8	394.1	12.8
	80 ft/sec	506.6	491.8	32.4
	100 ft/sec	634.5	634.5	26.8
10 sec	40 ft/sec	537.9	502.4	138.5
	60 ft/sec	533.1	470.6	116.1
	80 ft/sec	714.1	681.9	98.7
	100 ft/sec	737.7	749.3	61.7
15 sec	40 ft/sec	736.5	674.9	100.5
	60 ft/sec	831.0	845.5	130.8
	80 ft/sec	1044.1	1140.9	234.4
	100 ft/sec	1149.7	1049.7	384.9
20 sec	40 ft/sec	748.9	753.6	186.9
	60 ft/sec	739.8	679.9	156.8
	80 ft/sec	779.2	747.1	132.5
	100 ft/sec	784.7	778.6	84.3

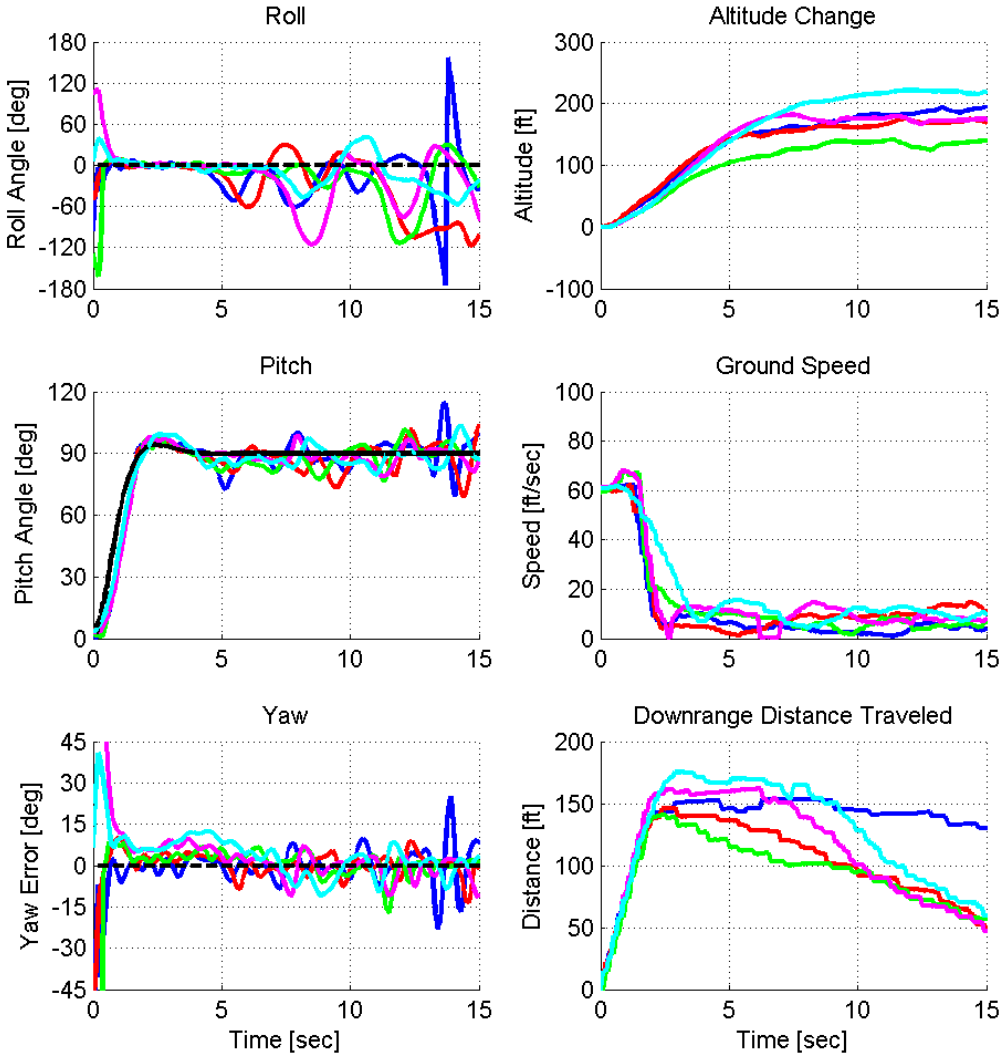
Flight Test: PID Reference Model Response, Rise Time = 1 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	82.0	90.7
2	0.0	yes	81.4	100.1
3	0.0	yes	59.9	94.4
4	0.0	yes	95.2	97.6
5	3.9	yes	7.2	94.6
	Average		78.1	95.5
	Median		81.4	94.6
	Std. Dev.		13.1	3.5



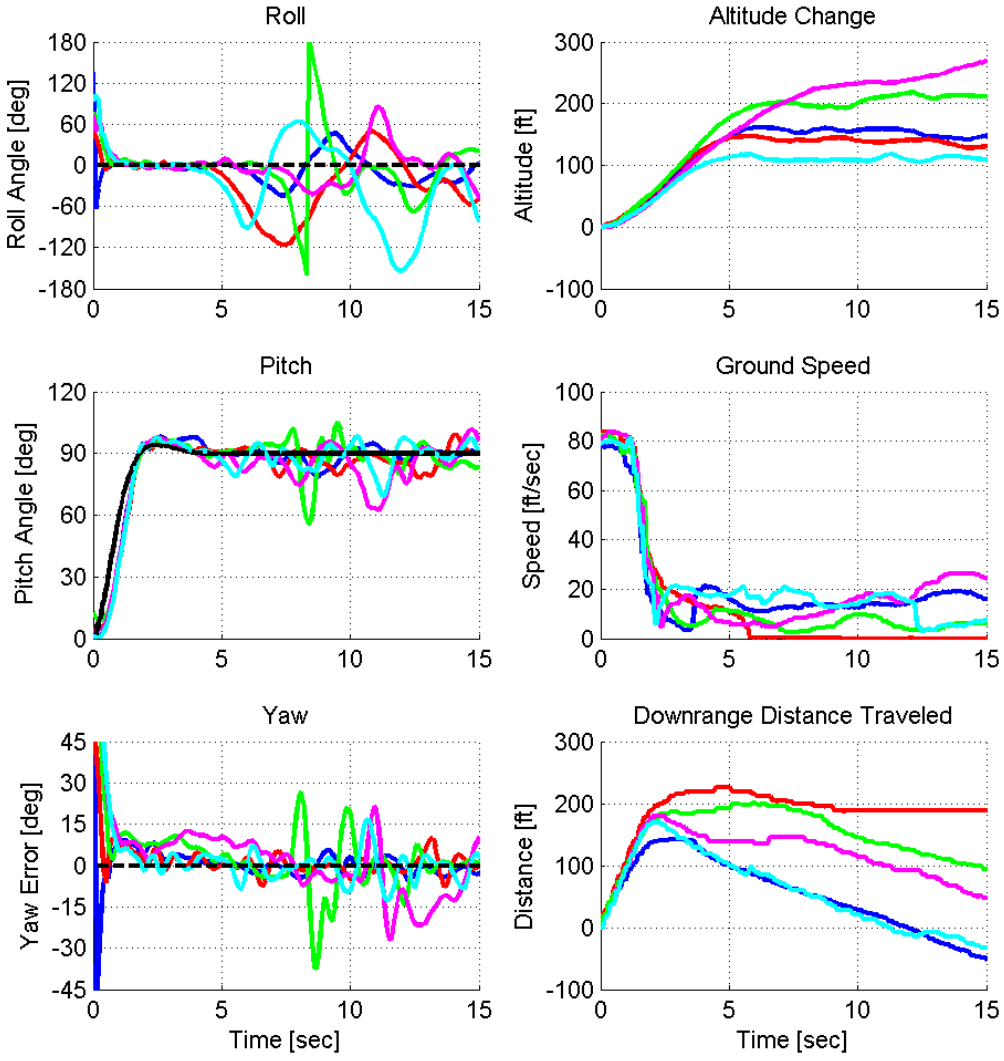
Flight Test: PID Reference Model Response, Rise Time = 1 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	8.5	yes	194.4	153.4
2	0.0	yes	175.6	146.2
3	8.5	yes	141.6	140.6
4	4.6	yes	182.0	161.7
5	3.0	yes	221.8	175.1
	Average		183.1	155.4
	Median		182.0	153.4
	Std. Dev.		29.2	13.5



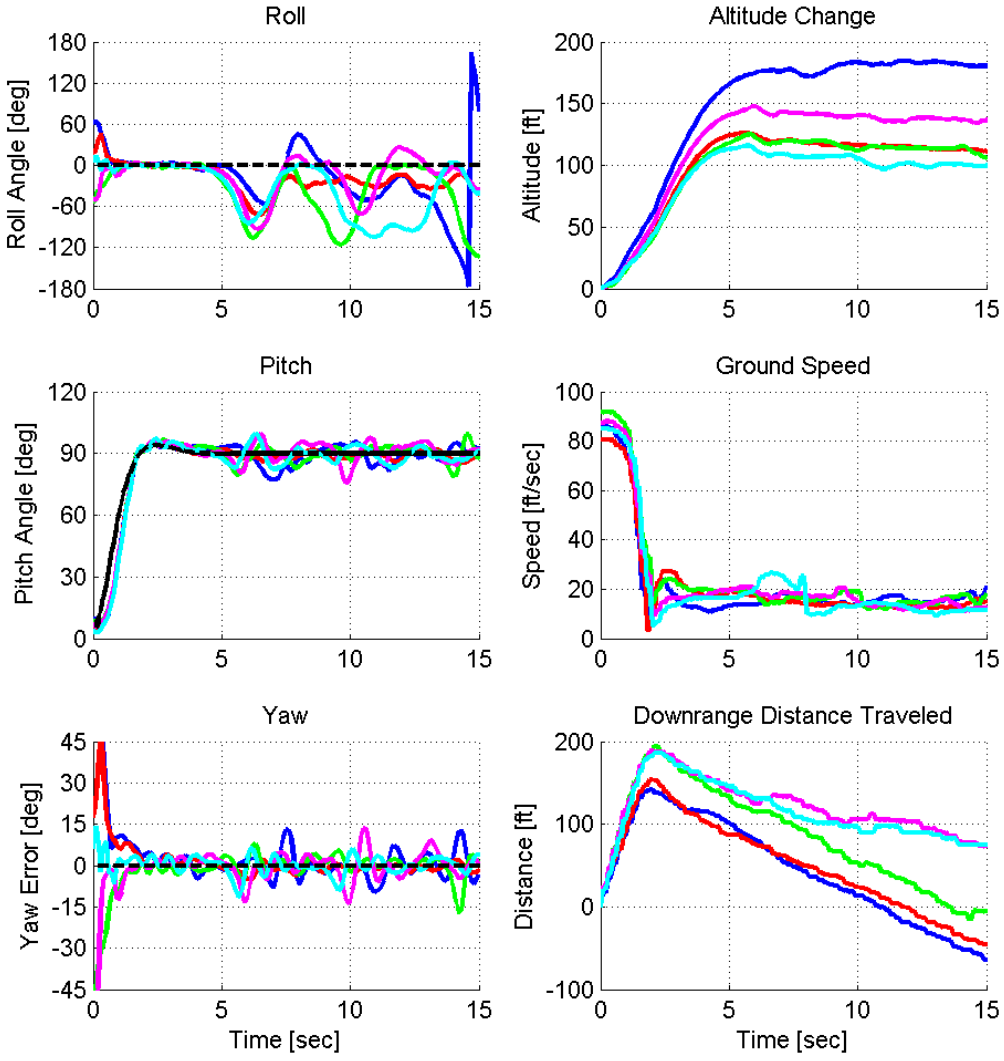
Flight Test: PID Reference Model Response, Rise Time = 1 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	161.4	142.3
2	1.3	yes	147.3	226.0
3	4.6	yes	218.0	200.5
4	0.0	yes	269.7	179.5
5	6.2	yes	118.1	168.8
	Average		182.9	183.4
	Median		161.4	179.5
	Std. Dev.		60.6	31.7



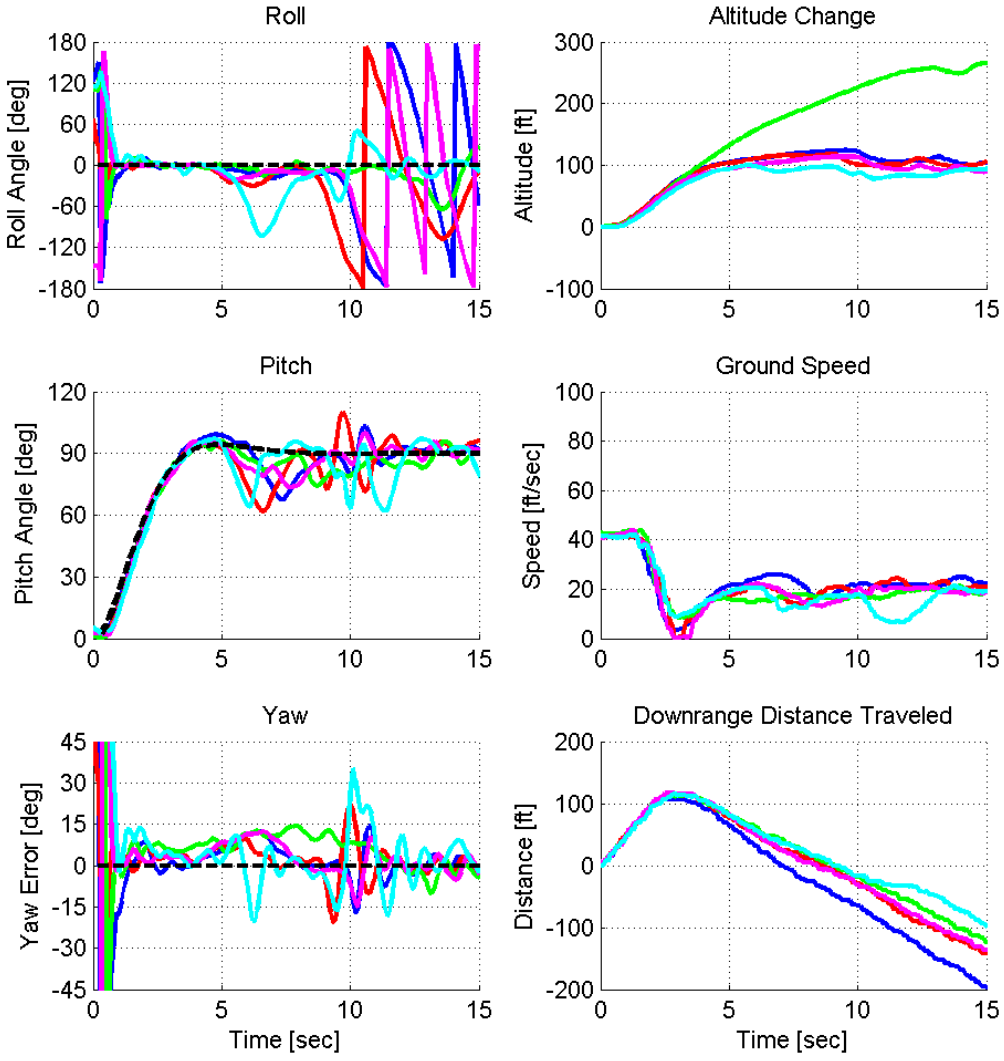
Flight Test: PID Reference Model Response, Rise Time = 1 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	184.7	141.7
2	0.0	yes	126.0	153.5
3	0.0	yes	125.1	194.1
4	2.3	yes	147.7	188.1
5	6.6	yes	115.9	185.8
	Average		139.9	172.6
	Median		126.0	185.8
	Std. Dev.		27.6	23.4



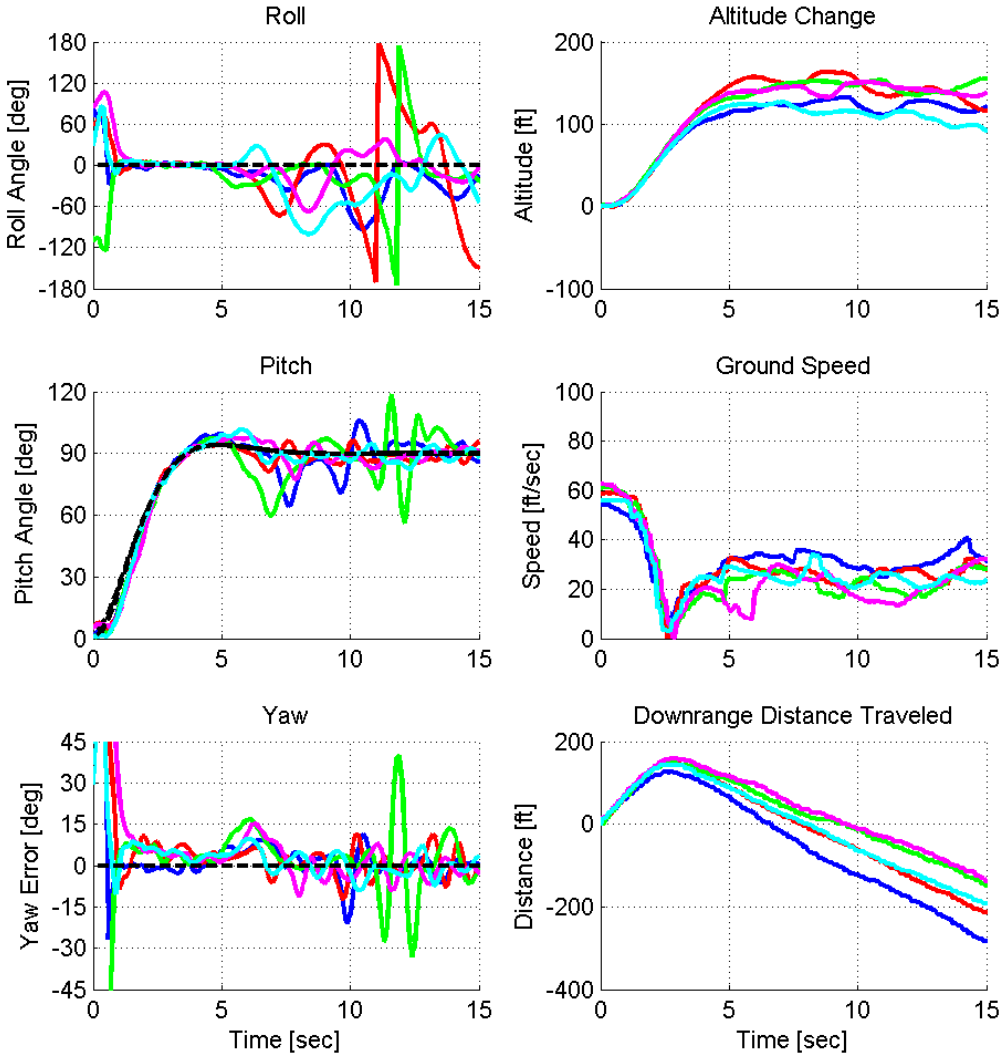
Flight Test: PID Reference Model Response, Rise Time = 2 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	4.6	yes	124.4	105.5
2	12.5	yes	117.8	115.5
3	16.4	yes	265.6	113.0
4	4.9	yes	114.3	116.4
5	6.2	yes	100.5	115.3
	Average		144.5	113.2
	Median		117.8	115.3
	Std. Dev.		68.3	4.5



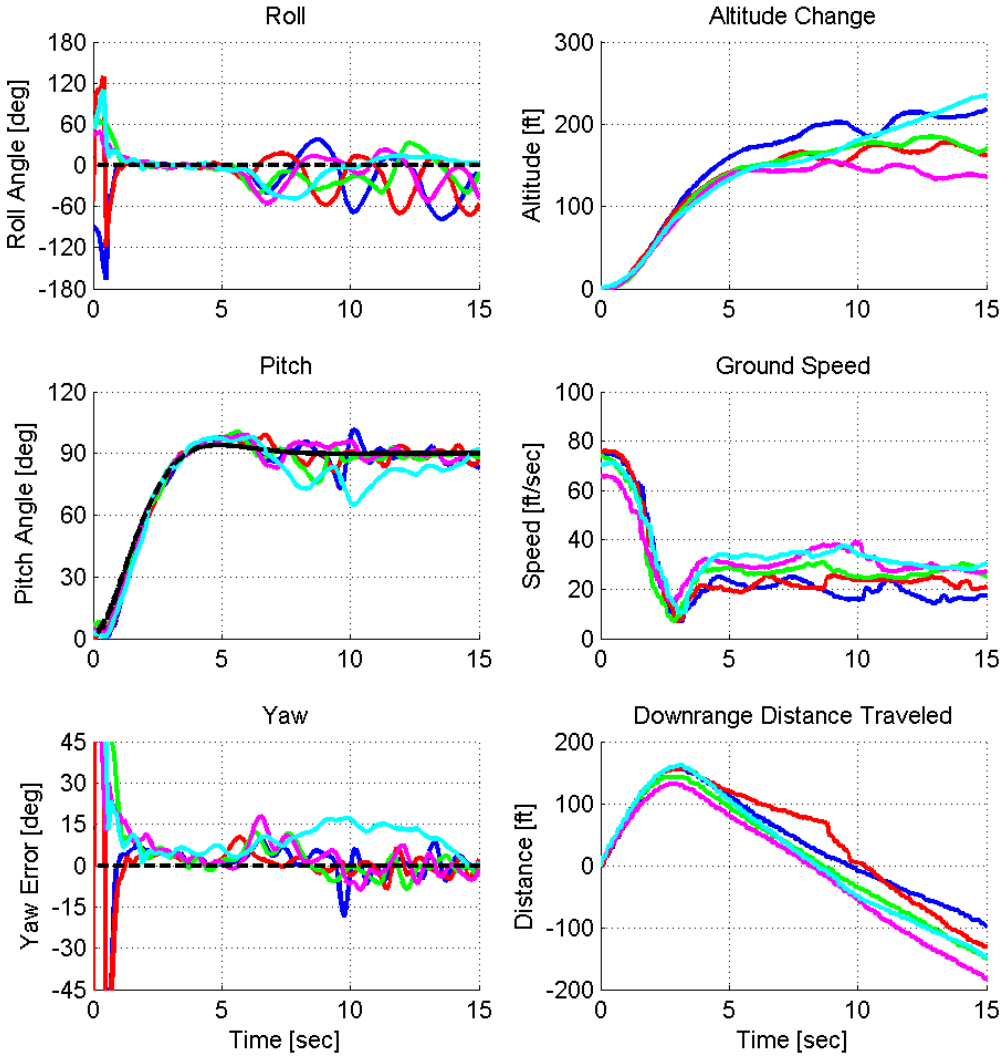
Flight Test: PID Reference Model Response, Rise Time = 2 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	5.2	yes	132.6	126.1
2	3.9	yes	163.6	149.3
3	9.8	yes	155.6	149.8
4	13.8	yes	151.1	157.5
5	9.2	yes	126.8	142.0
	Average		145.9	144.9
	Median		151.1	149.3
	Std. Dev.		15.6	11.9



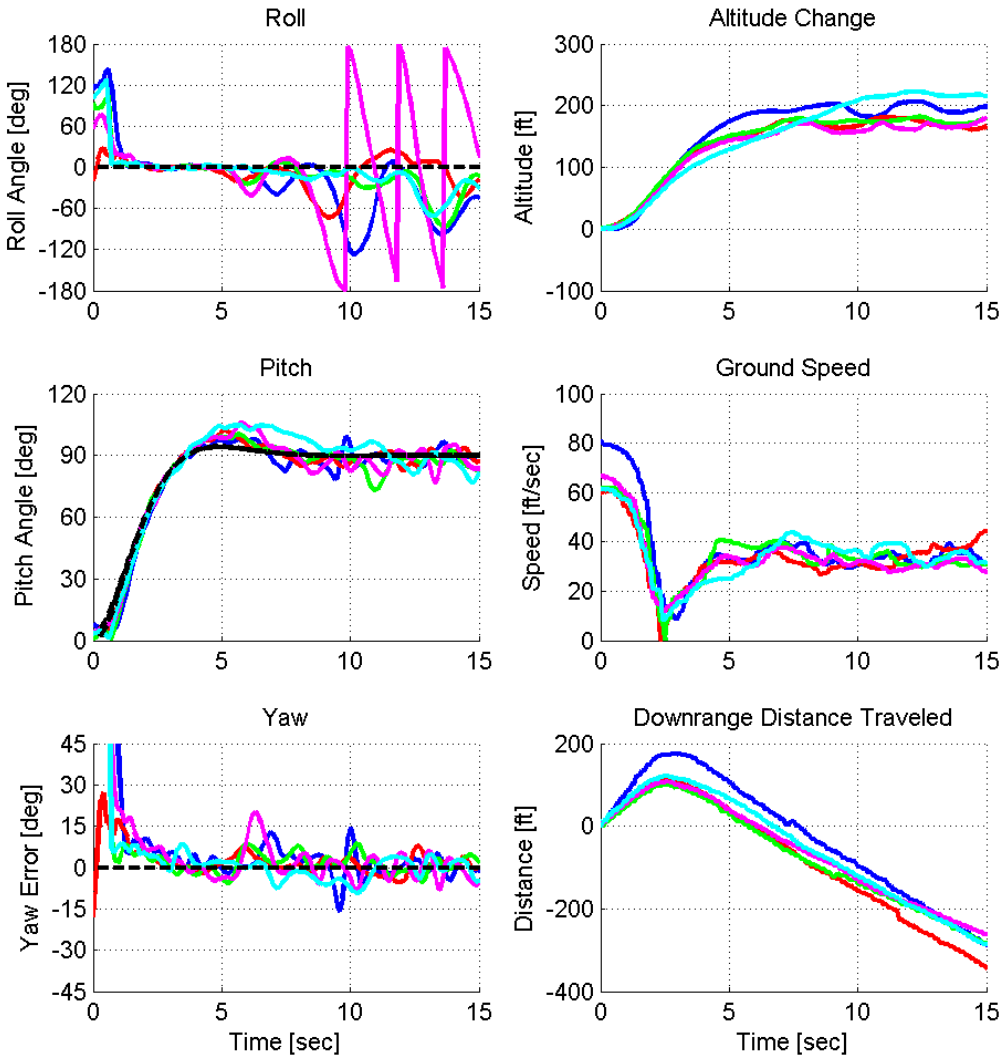
Flight Test: PID Reference Model Response, Rise Time = 2 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	7.2	yes	217.5	156.4
2	12.5	yes	178.2	156.3
3	7.5	yes	184.5	142.7
4	11.2	yes	154.3	131.1
5	13.8	yes	234.6	161.1
	Average		193.8	149.5
	Median		184.5	156.3
	Std. Dev.		32.1	12.4



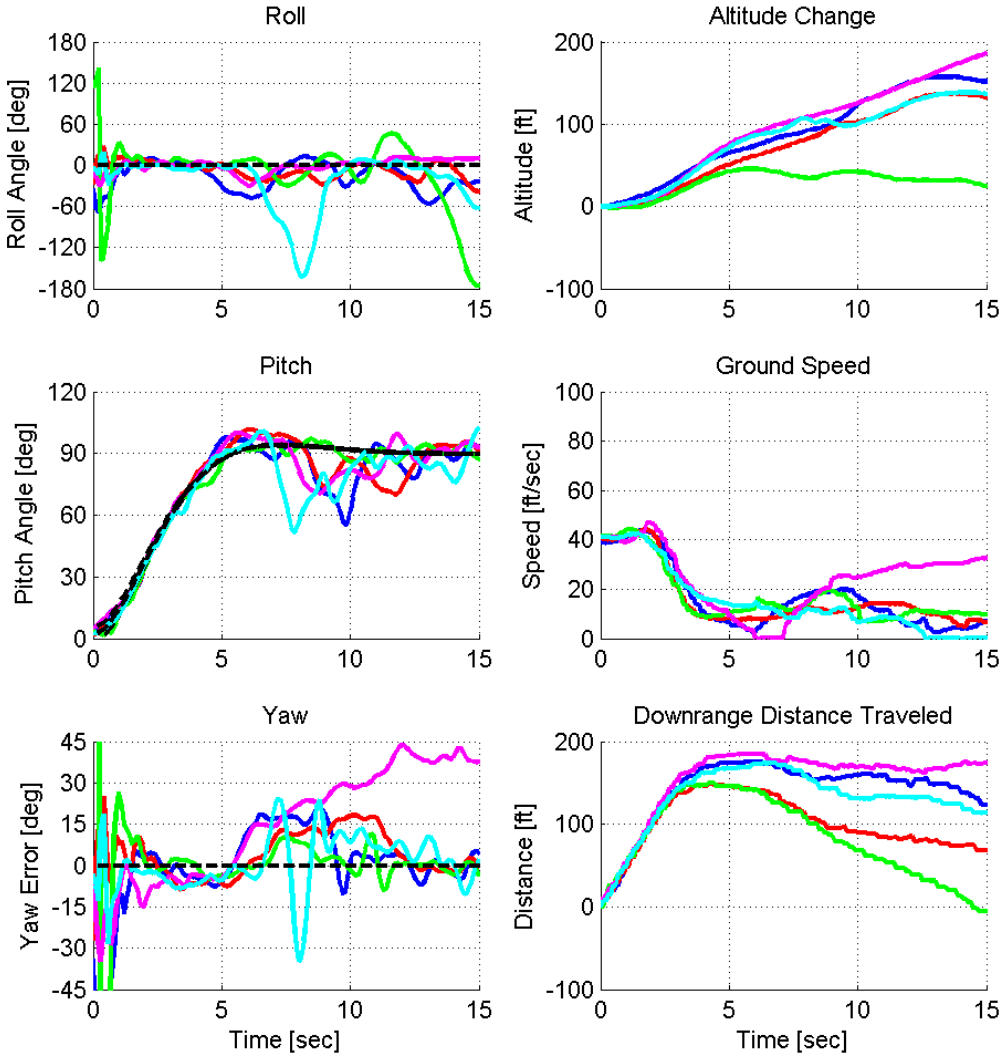
Flight Test: PID Reference Model Response, Rise Time = 2 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	11.2	yes	206.7	174.9
2	3.9	yes	181.5	111.4
3	7.5	yes	181.7	98.4
4	6.9	yes	180.7	106.4
5	5.9	yes	222.5	121.5
	Average		194.6	122.5
	Median		181.7	111.4
	Std. Dev.		19.1	30.4



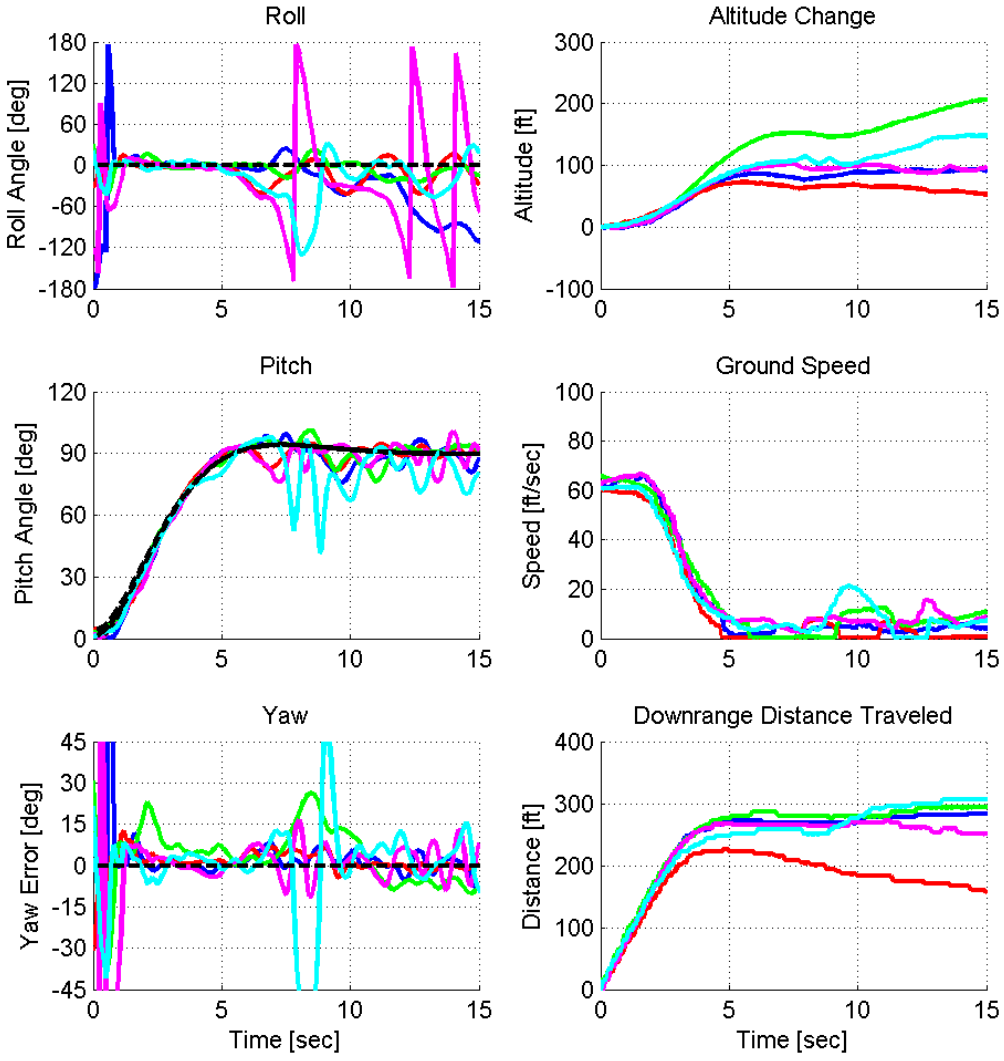
Flight Test: PID Reference Model Response, Rise Time = 3 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	157.6	175.2
2	0.0	yes	137.2	148.4
3	8.9	yes	45.8	150.1
4	3.0	no	186.4	184.9
5	8.5	yes	138.7	174.2
	Average		119.8	162.0
	Median		138.0	162.2
	Std. Dev.		50.2	14.7



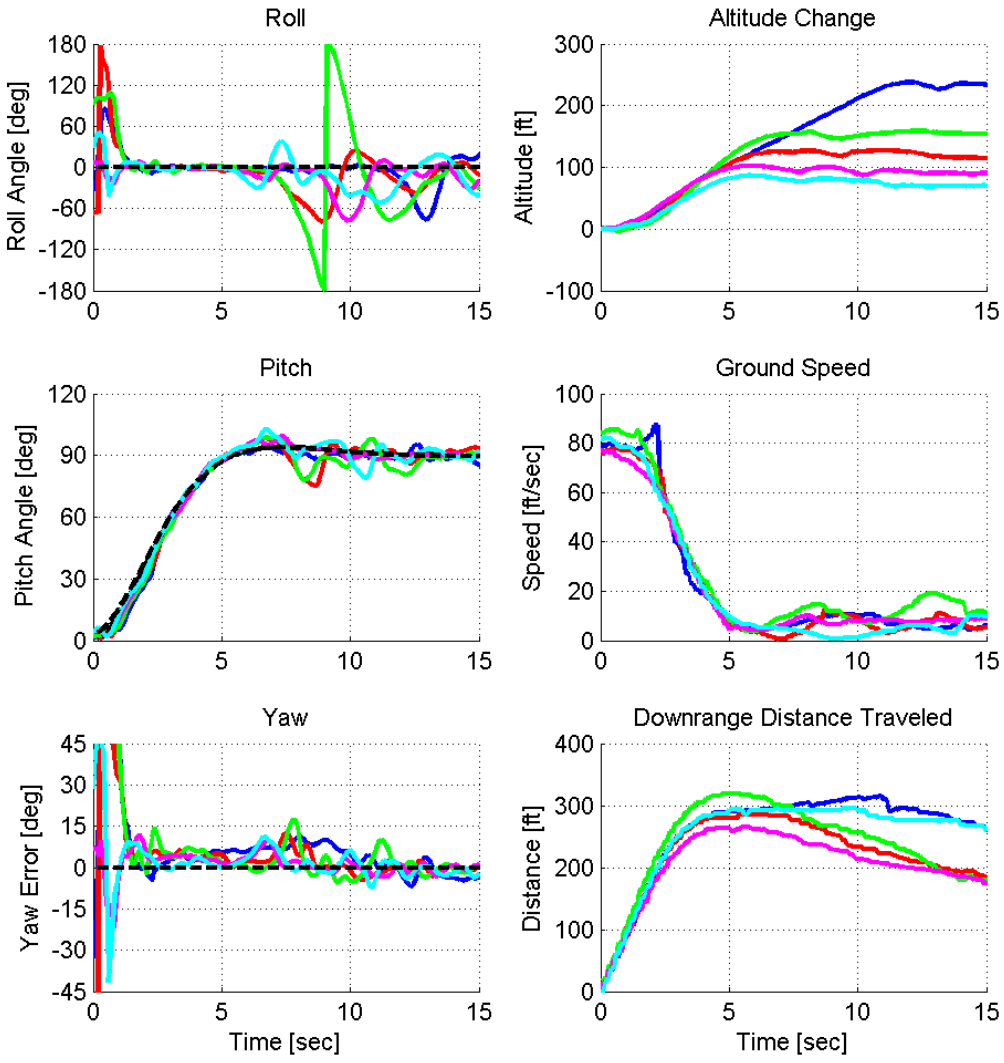
Flight Test: PID Reference Model Response, Rise Time = 3 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	4.6	yes	93.7	282.2
2	0.0	yes	72.3	226.0
3	0.0	yes	206.9	293.8
4	4.9	yes	102.5	272.9
5	0.0	yes	148.5	306.6
	Average		124.8	276.3
	Median		102.5	282.2
	Std. Dev.		53.6	30.8



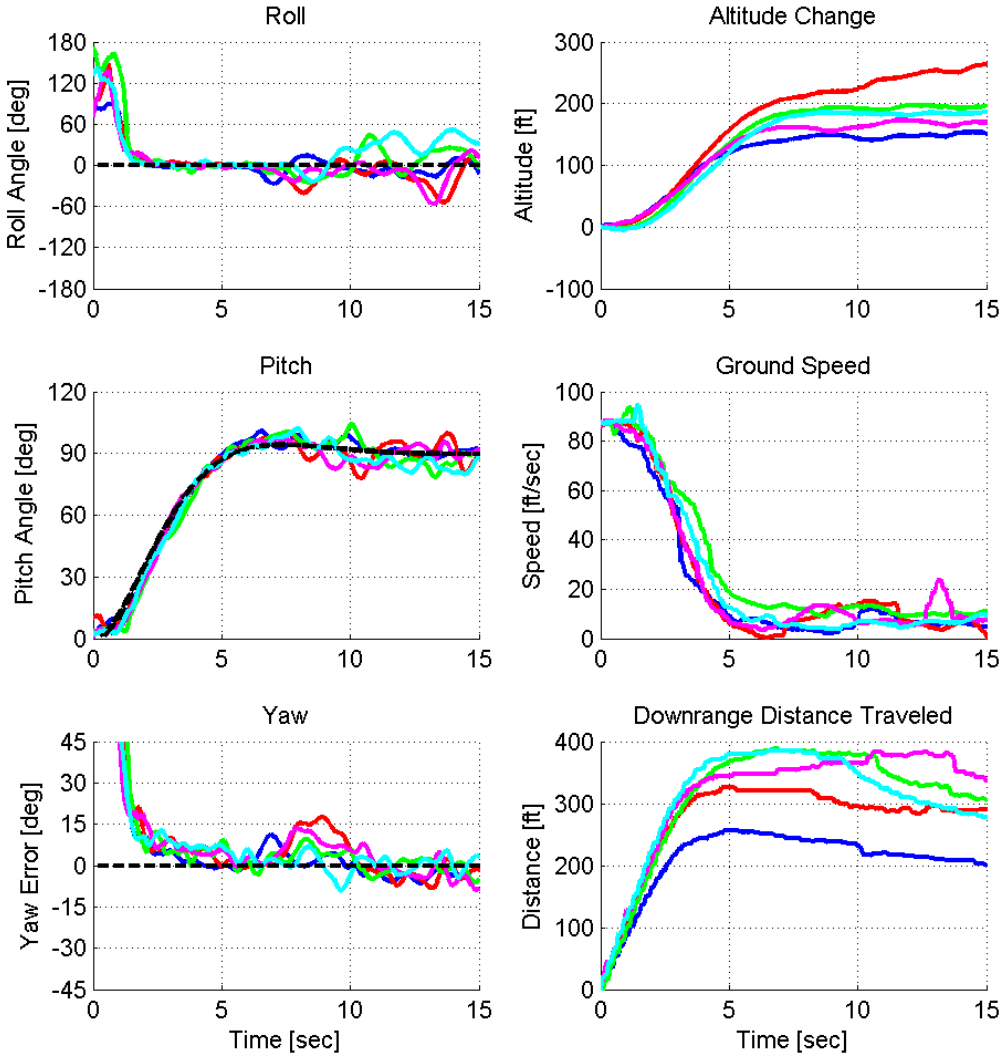
Flight Test: PID Reference Model Response, Rise Time = 3 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	8.2	yes	238.0	314.5
2	15.1	yes	127.7	285.4
3	6.6	yes	158.9	318.2
4	9.8	yes	102.7	264.4
5	7.9	yes	86.6	295.2
	Average		142.8	295.5
	Median		127.7	295.2
	Std. Dev.		59.8	22.1



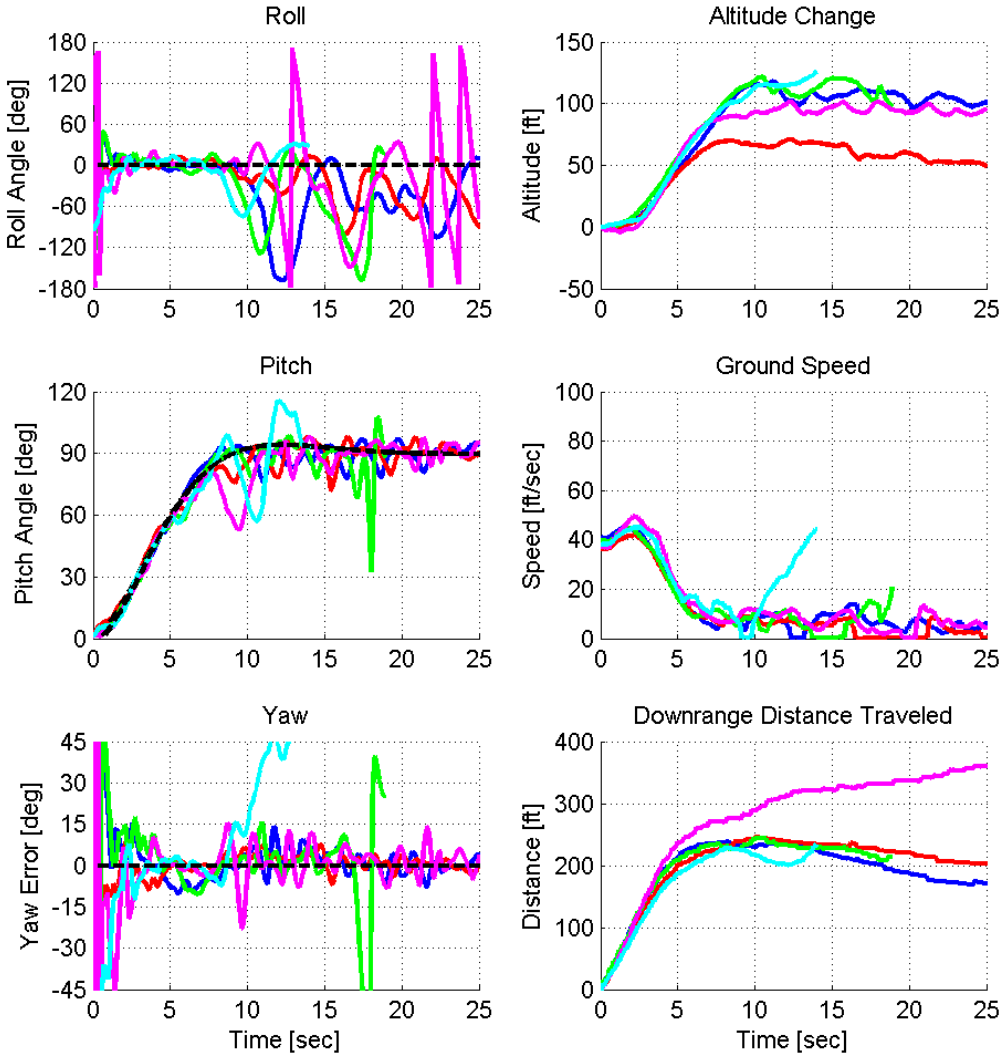
Flight Test: PID Reference Model Response, Rise Time = 3 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	4.3	yes	153.6	255.7
2	2.0	yes	263.5	326.5
3	2.3	yes	197.6	388.0
4	7.2	yes	173.0	383.8
5	2.3	yes	186.9	384.3
	Average		195.0	347.7
	Median		186.9	383.8
	Std. Dev.		41.7	57.4



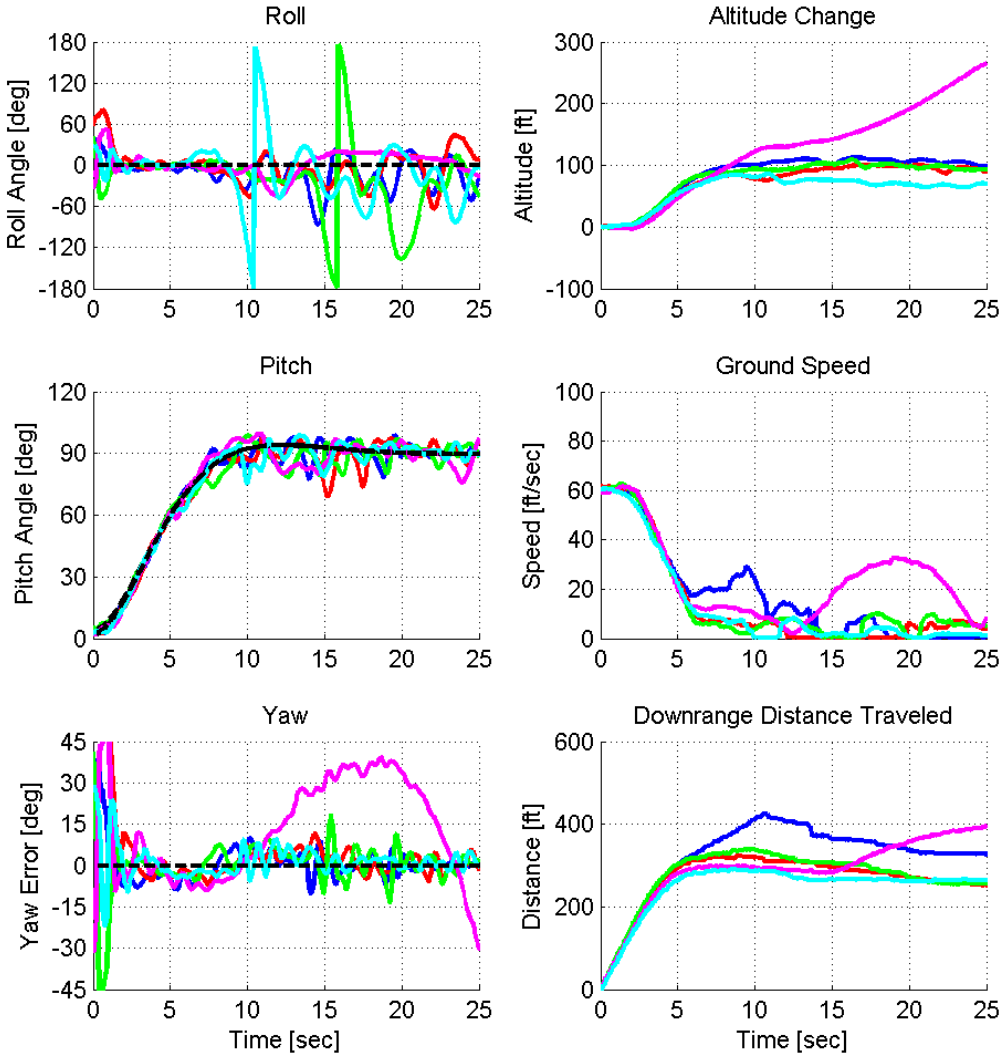
Flight Test: PID Reference Model Response, Rise Time = 5 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	8.5	yes	117.9	239.9
2	3.6	yes	71.1	245.0
3	1.0	no	121.7	243.0
4	1.0	yes	101.4	359.8
5	3.6	no	125.4	233.9
	Average		96.8	281.6
	Median		101.4	245.0
	Std. Dev.		23.7	67.8



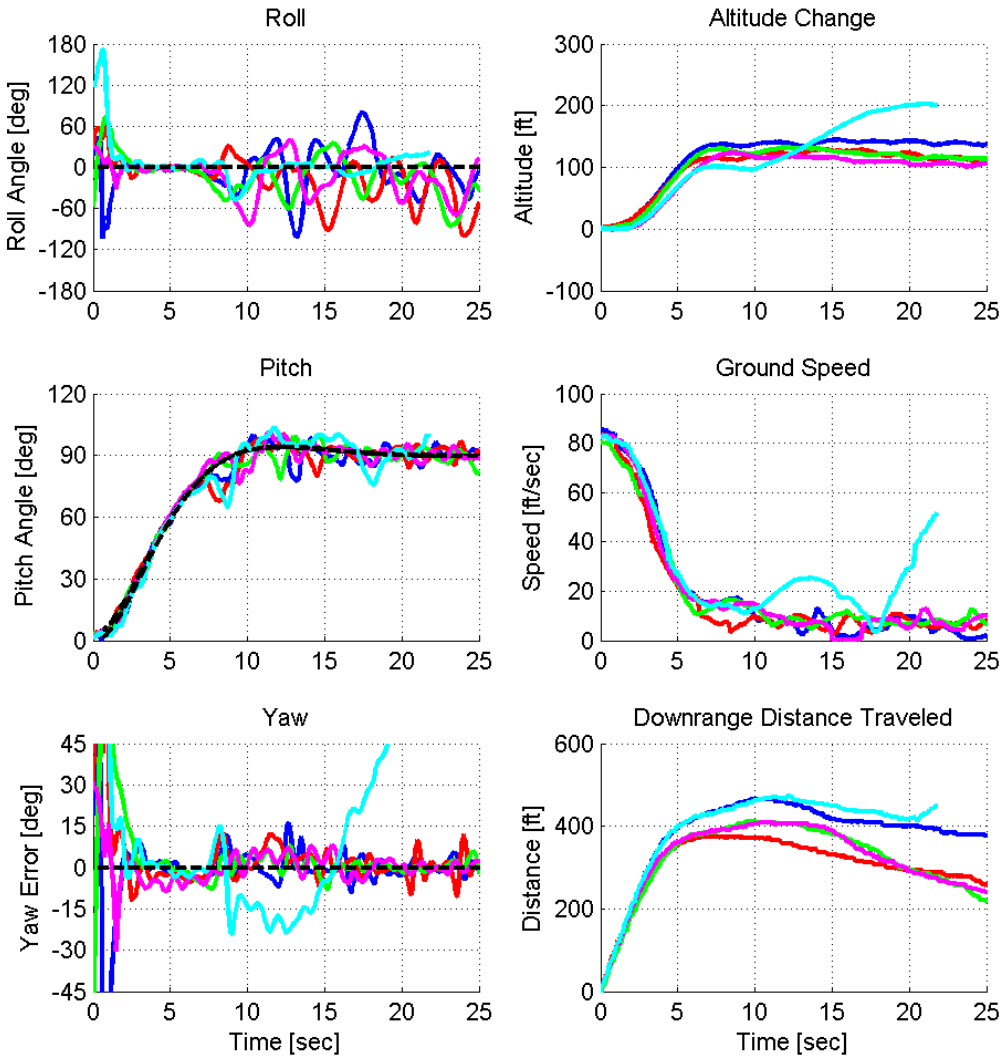
Flight Test: PID Reference Model Response, Rise Time = 5 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	112.2	424.1
2	3.3	yes	100.9	322.6
3	2.3	yes	109.2	338.7
4	1.6	no	265.0	396.8
5	0.0	yes	86.6	291.6
	Average		102.2	344.2
	Median		105.1	330.7
	Std. Dev.		11.5	56.7



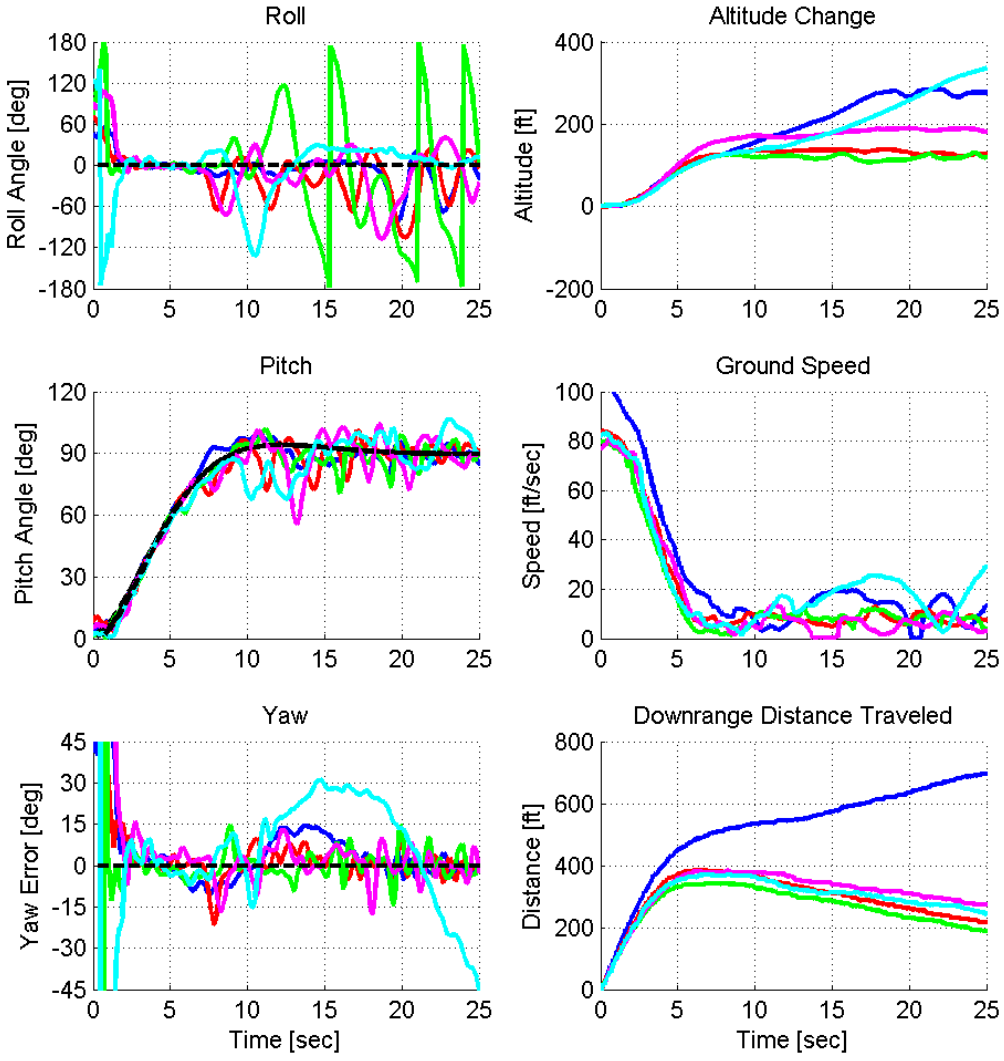
Flight Test: PID Reference Model Response, Rise Time = 5 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	2.0	yes	144.6	466.0
2	0.0	yes	128.9	375.2
3	3.9	yes	132.1	412.5
4	6.2	yes	123.4	408.1
5	7.9	no	202.6	472.0
	Average		132.3	415.4
	Median		130.5	410.3
	Std. Dev.		9.0	37.5



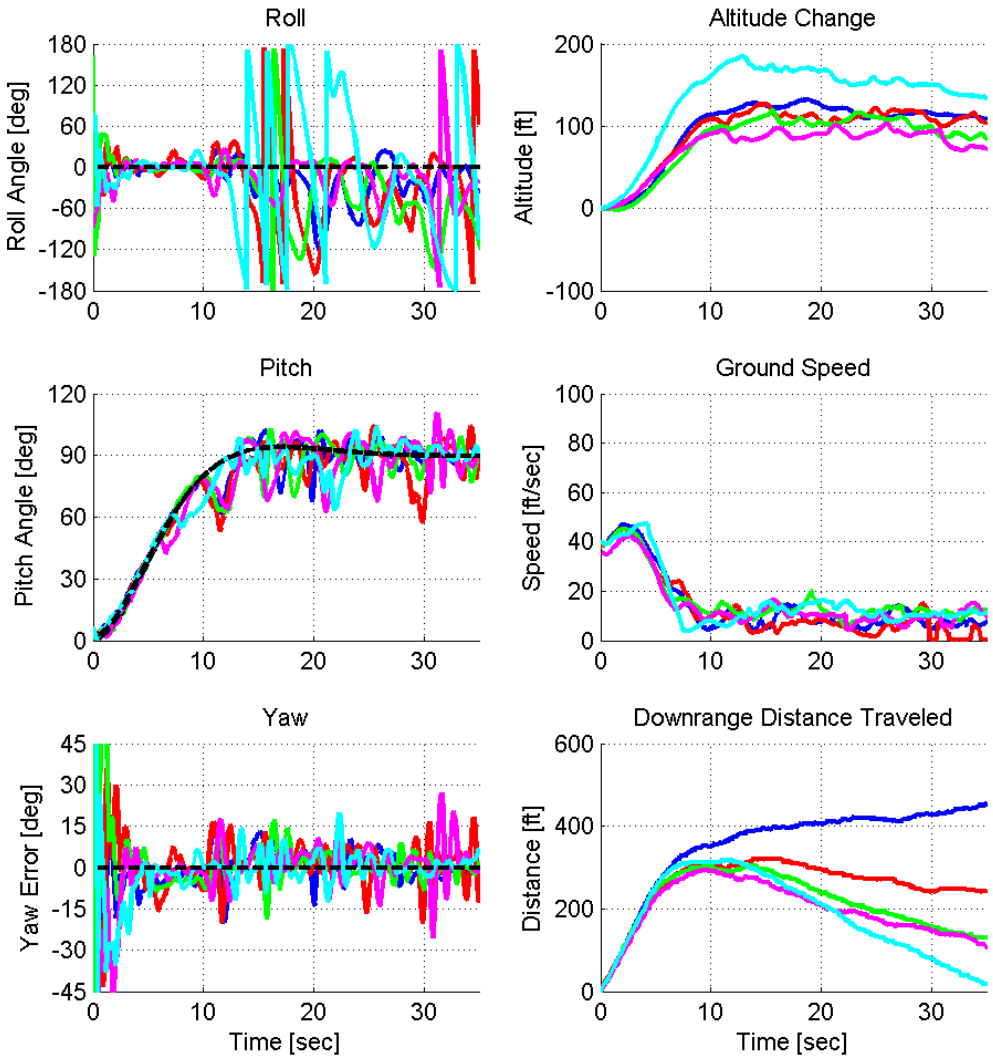
Flight Test: PID Reference Model Response, Rise Time = 5 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	7.9	yes	285.1	695.8
2	8.2	yes	138.6	382.8
3	6.9	yes	130.0	342.1
4	3.3	yes	190.2	383.7
5	6.2	yes	336.0	369.6
	Average		216.0	434.8
	Median		190.2	382.8
	Std. Dev.		91.1	146.8



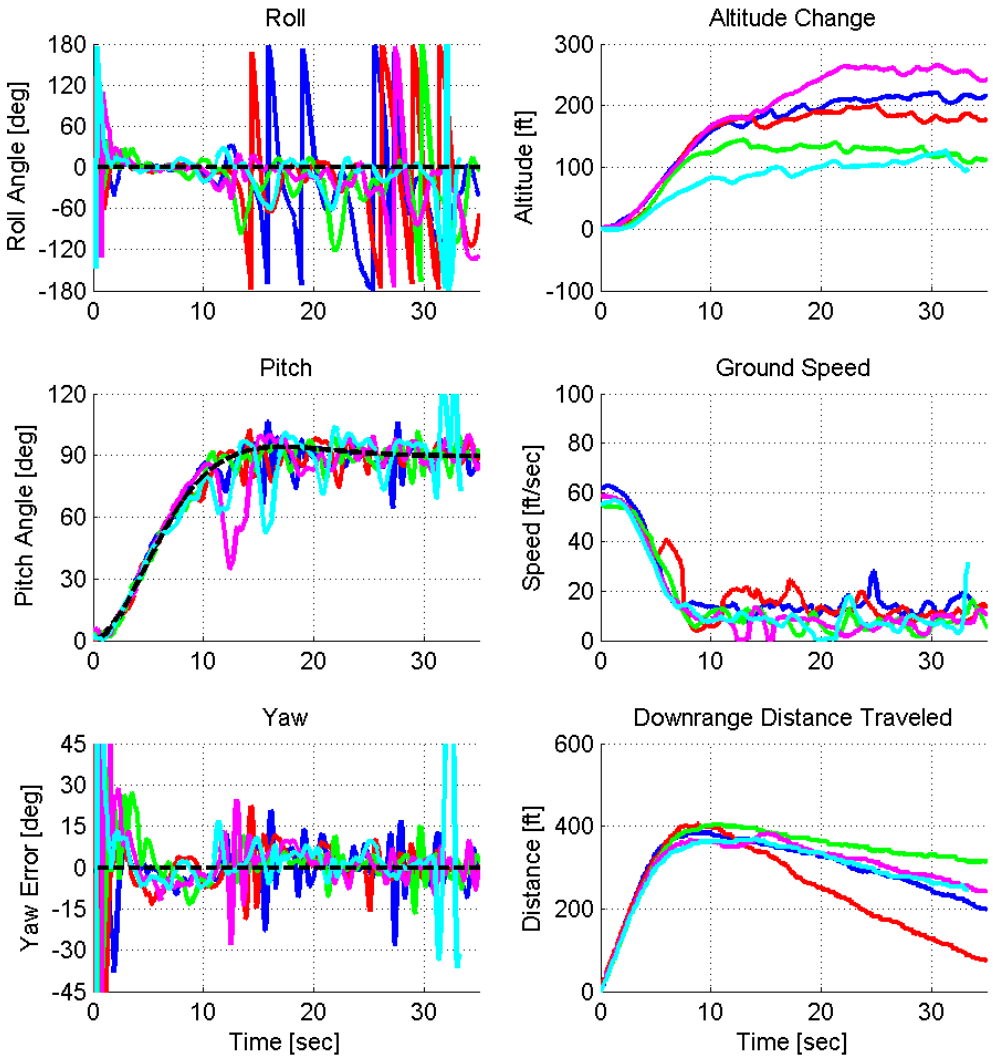
Flight Test: PID Reference Model Response, Rise Time = 7 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	4.6	yes	132.2	452.9
2	6.2	yes	127.3	320.5
3	7.5	yes	119.5	304.1
4	3.6	yes	103.1	293.8
5	2.6	yes	185.8	316.5
	Average		133.6	337.6
	Median		127.3	316.5
	Std. Dev.		31.2	65.3



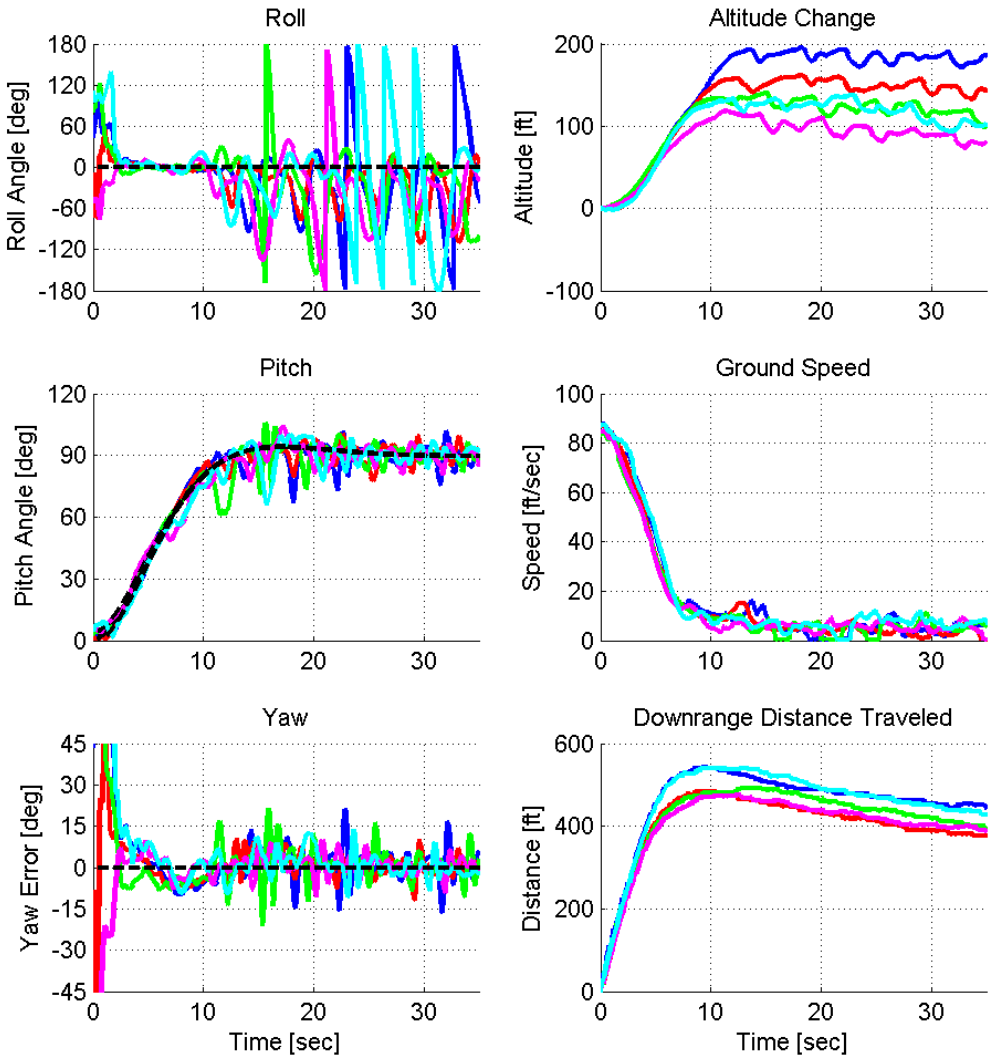
Flight Test: PID Reference Model Response, Rise Time = 7 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	1.0	yes	220.7	384.1
2	5.2	yes	201.3	405.6
3	0.0	yes	144.5	404.1
4	0.0	yes	265.2	381.6
5	0.0	no	127.4	367.5
	Average		207.9	393.8
	Median		211.0	394.1
	Std. Dev.		50.0	12.8



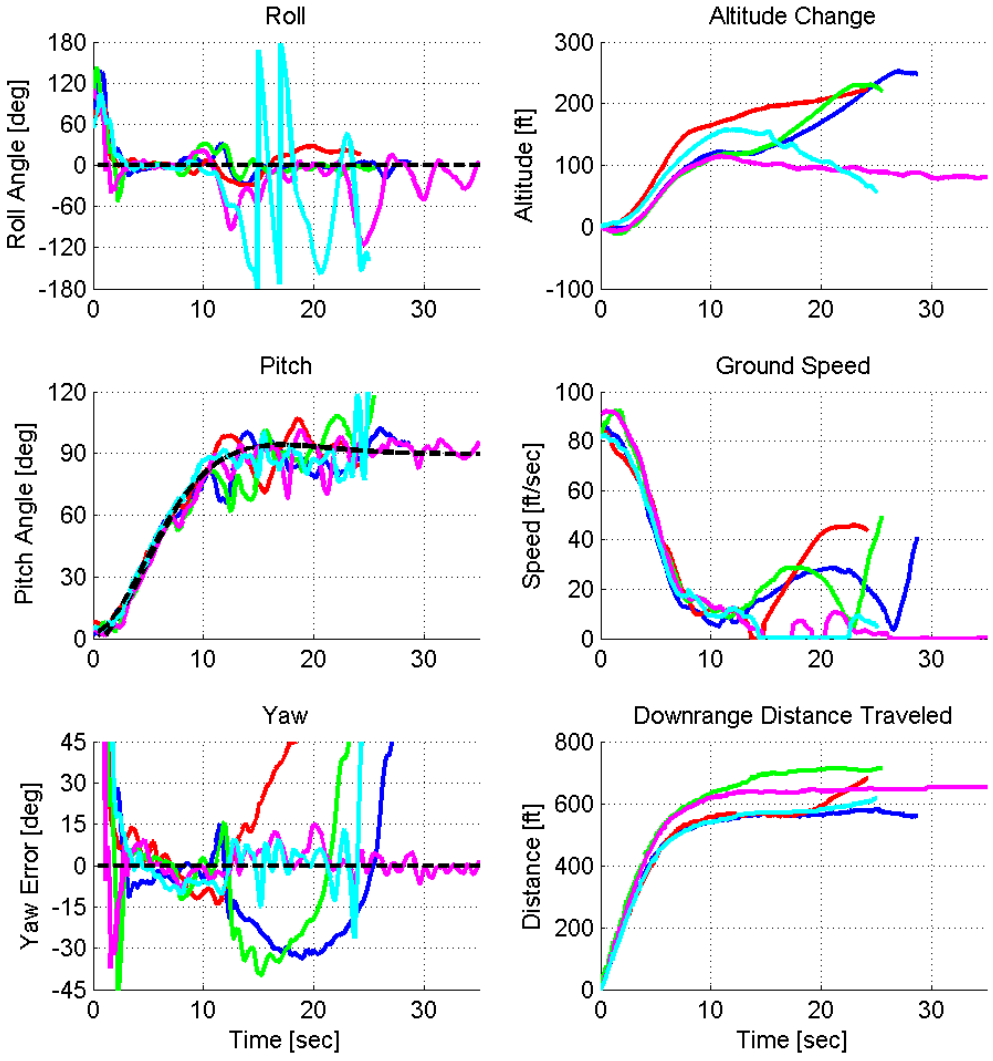
Flight Test: PID Reference Model Response, Rise Time = 7 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	195.7	541.7
2	3.0	yes	161.7	483.6
3	0.0	yes	140.3	491.8
4	0.0	yes	118.4	474.7
5	0.0	yes	137.9	541.1
	Average		150.8	506.6
	Median		140.3	491.8
	Std. Dev.		29.4	32.4



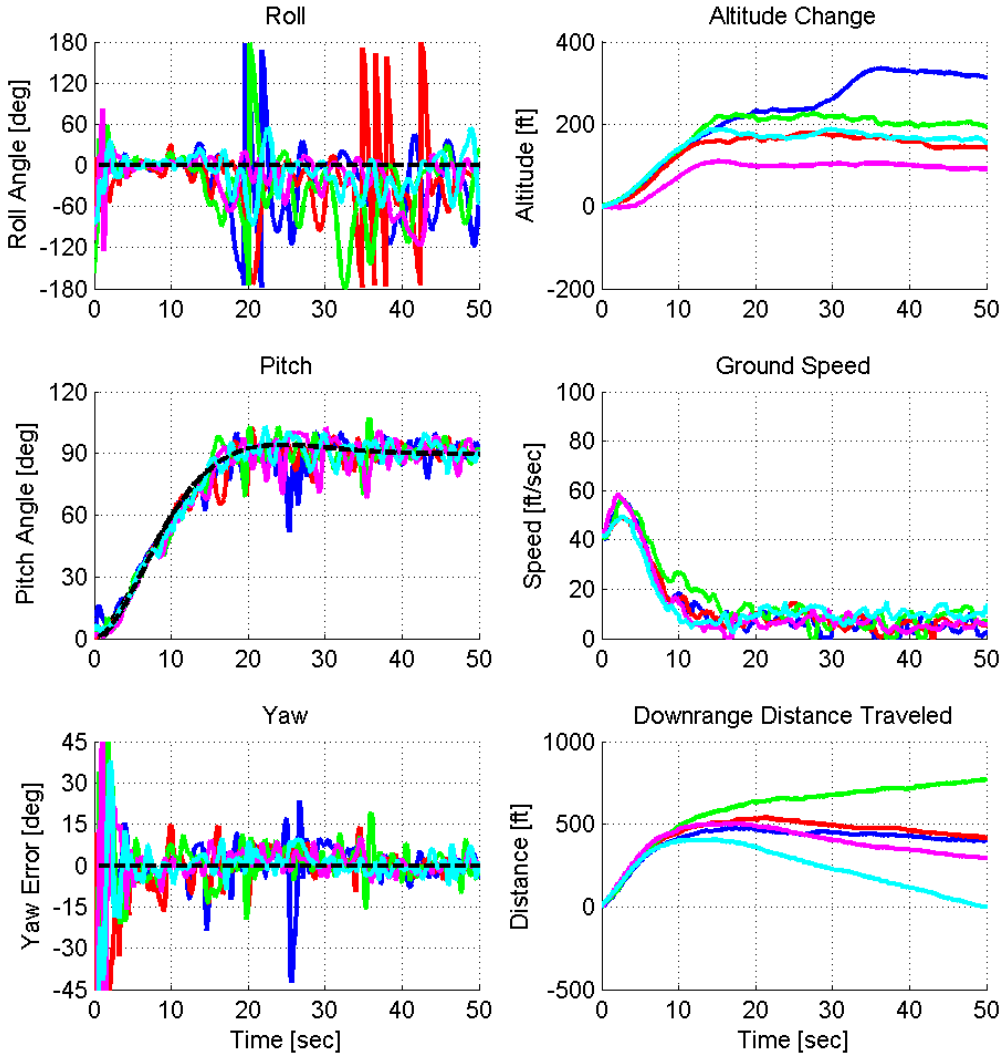
Flight Test: PID Reference Model Response, Rise Time = 7 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	251.9	581.5
2	1.0	no	223.8	679.5
3	0.0	no	230.2	714.1
4	3.3	yes	114.3	653.4
5	4.9	yes	157.4	615.5
	Average		135.8	634.5
	Median		135.8	634.5
	Std. Dev.		30.5	26.8



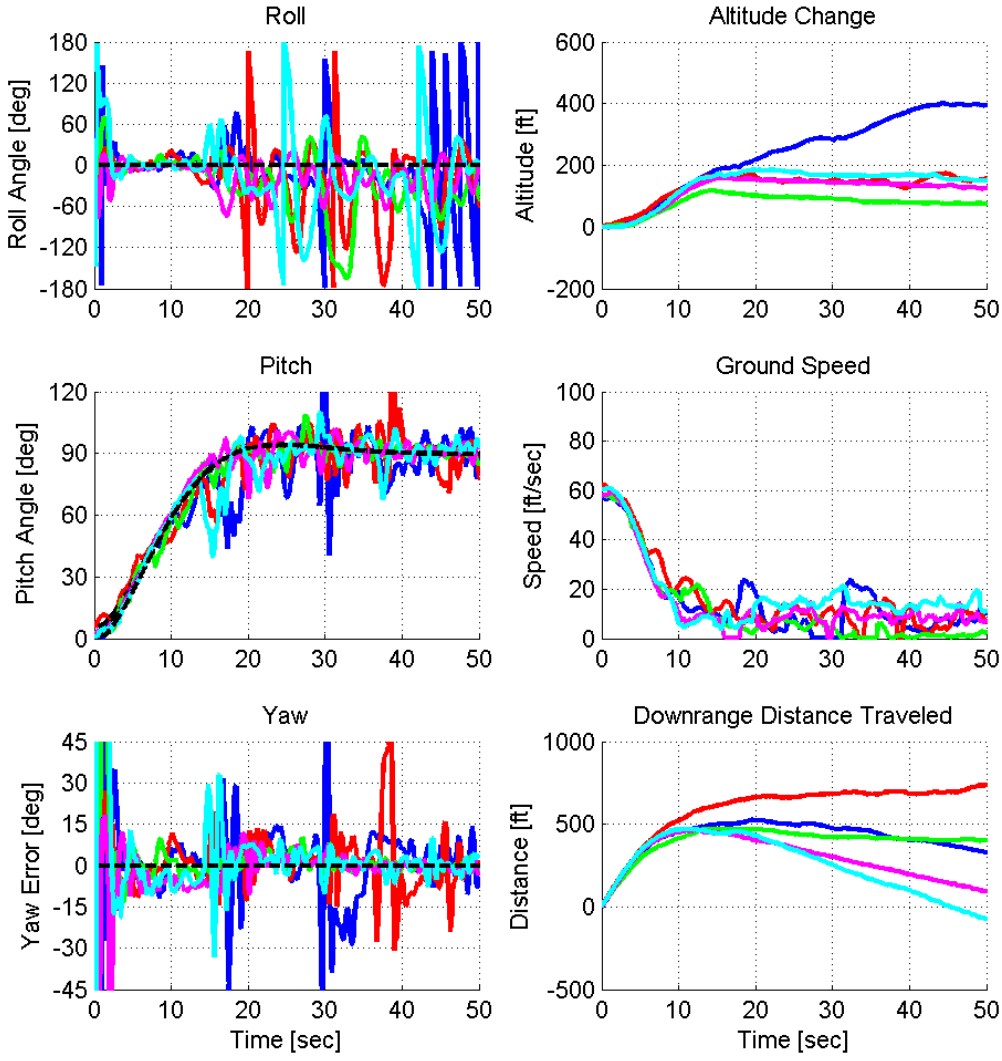
Flight Test: PID Reference Model Response, Rise Time = 10 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	7.5	yes	335.4	474.3
2	4.6	yes	178.2	538.2
3	8.5	yes	225.6	769.6
4	12.5	yes	109.8	502.4
5	9.5	yes	187.5	404.8
	Average		207.3	537.9
	Median		187.5	502.4
	Std. Dev.		82.9	138.5



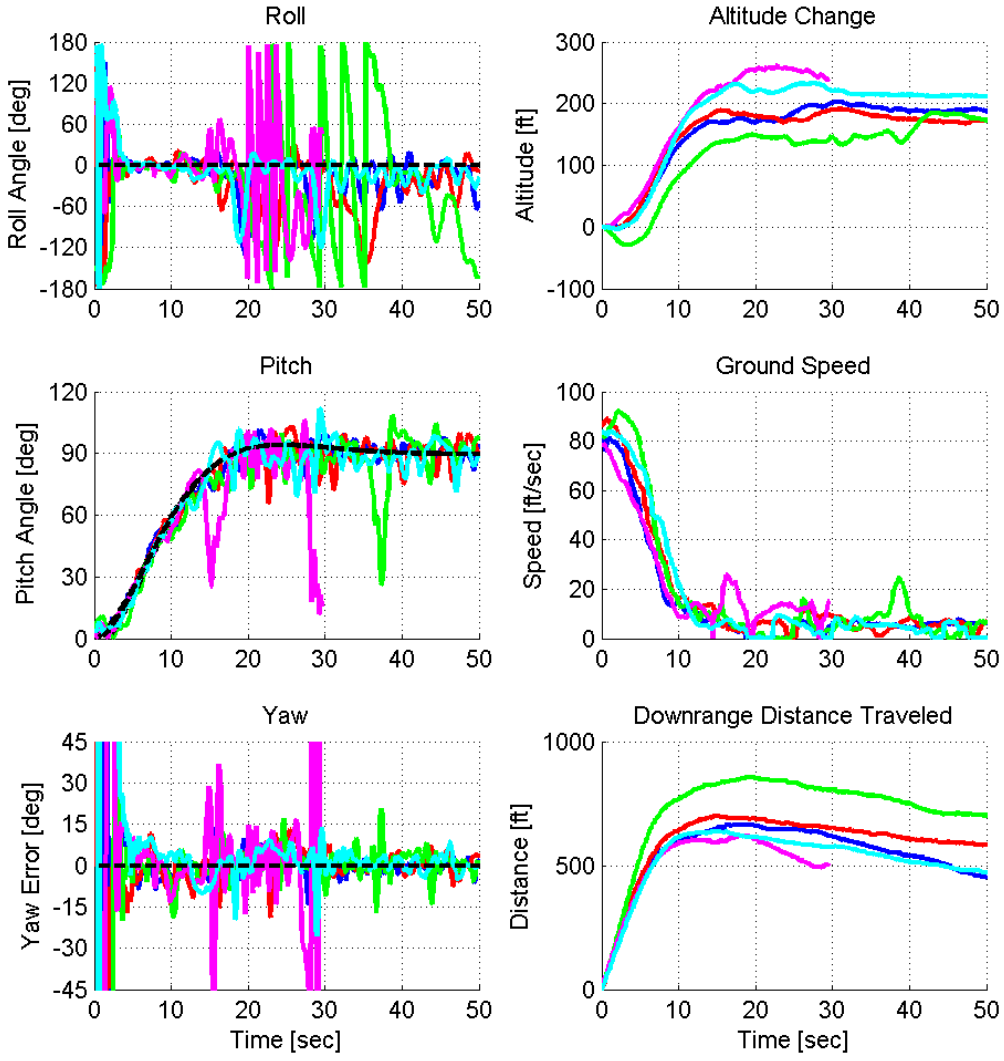
Flight Test: PID Reference Model Response, Rise Time = 10 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	1.6	yes	401.4	526.3
2	6.2	yes	173.5	735.8
3	9.2	yes	118.2	470.6
4	9.5	yes	158.7	463.6
5	13.8	yes	184.5	469.4
	Average		207.3	533.1
	Median		173.5	470.6
	Std. Dev.		111.4	116.1



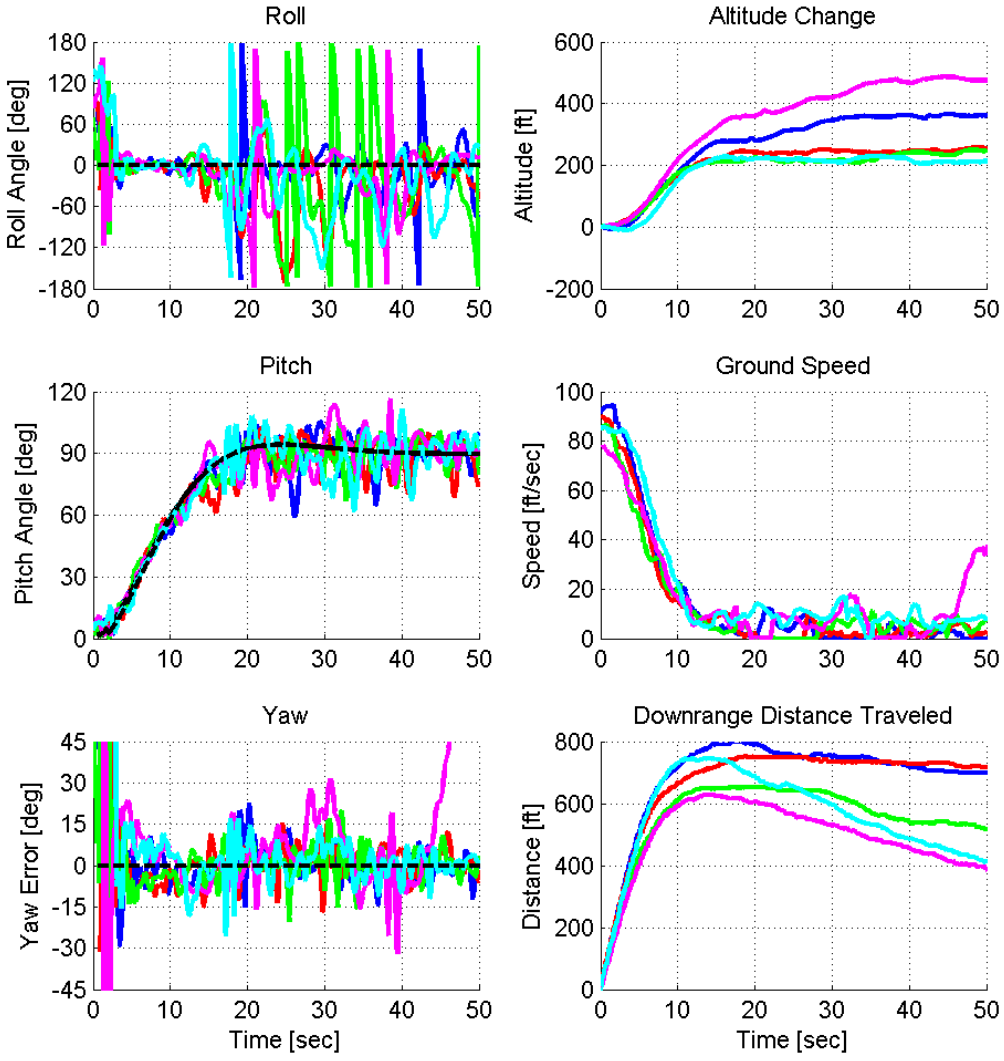
Flight Test: PID Reference Model Response, Rise Time = 10 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	8.9	yes	202.8	664.2
2	7.5	yes	191.0	699.5
3	7.5	yes	184.7	857.0
4	0.0	no	261.2	619.8
5	10.8	yes	233.1	635.7
	Average		202.9	714.1
	Median		196.9	681.9
	Std. Dev.		21.5	98.7



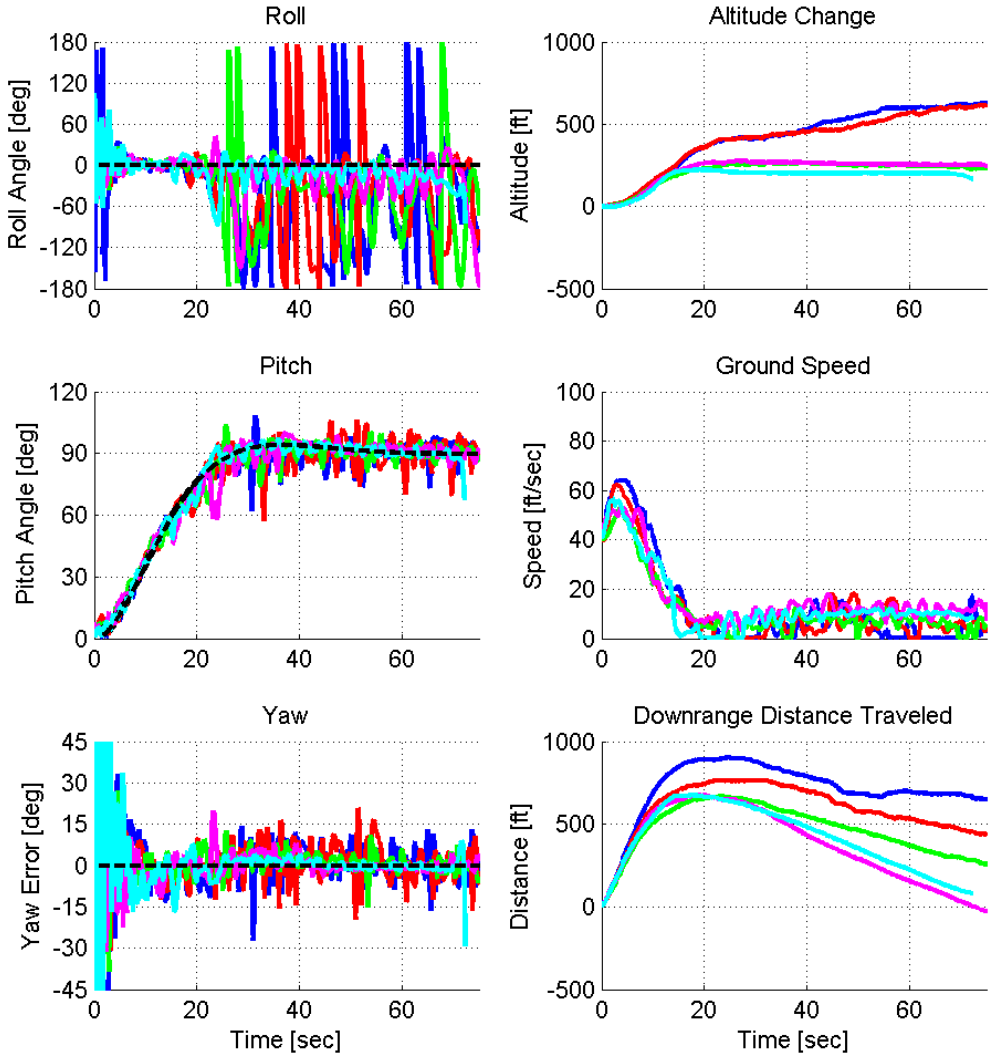
Flight Test: PID Reference Model Response, Rise Time = 10 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	4.3	yes	366.6	799.9
2	6.2	yes	258.7	752.1
3	6.9	yes	249.6	652.5
4	1.6	no	487.1	629.3
5	5.9	yes	230.2	746.4
	Average		276.3	737.7
	Median		254.2	749.3
	Std. Dev.		61.4	61.7



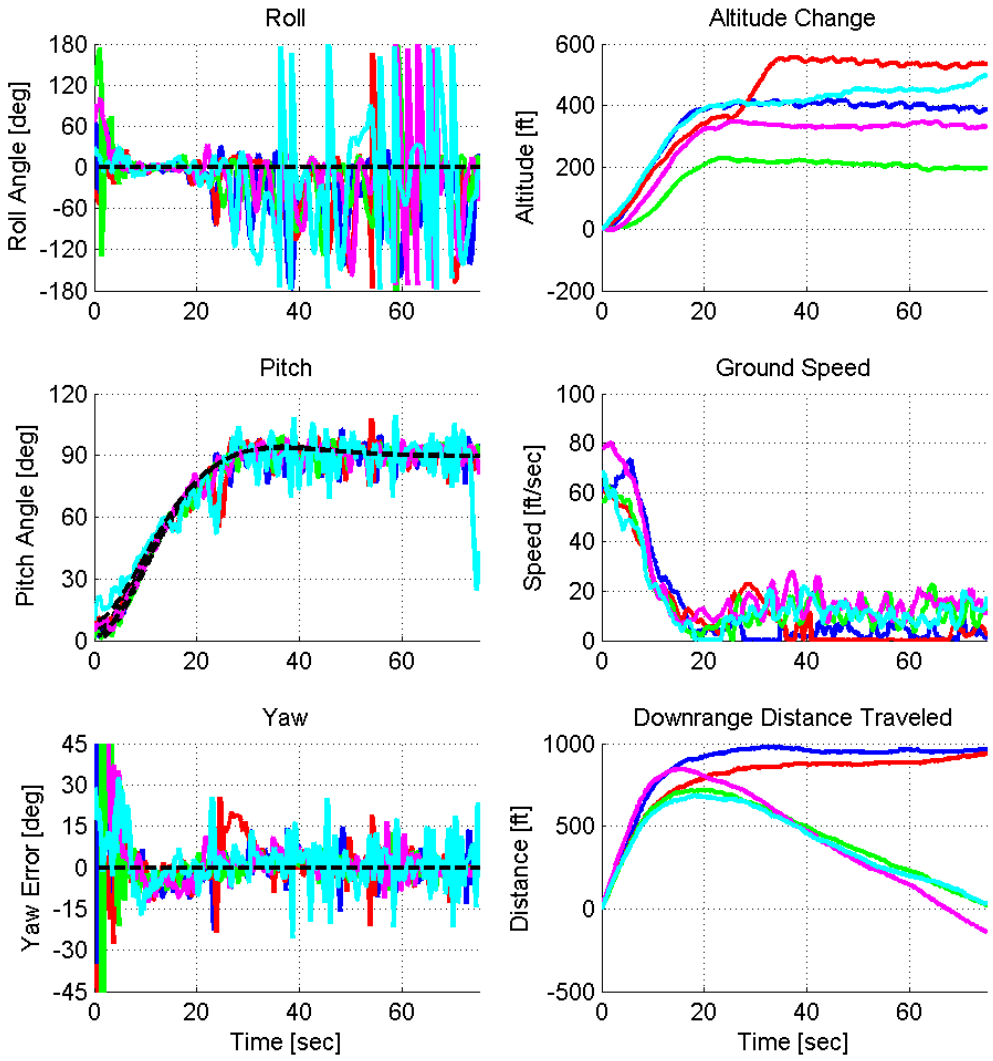
Flight Test: PID Reference Model Response, Rise Time = 15 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	7.5	yes	625.7	902.1
2	3.3	yes	616.2	762.8
3	7.5	yes	266.7	669.0
4	7.9	yes	278.1	674.9
5	4.9	yes	223.2	673.8
	Average		402.0	736.5
	Median		278.1	674.9
	Std. Dev.		200.9	100.5



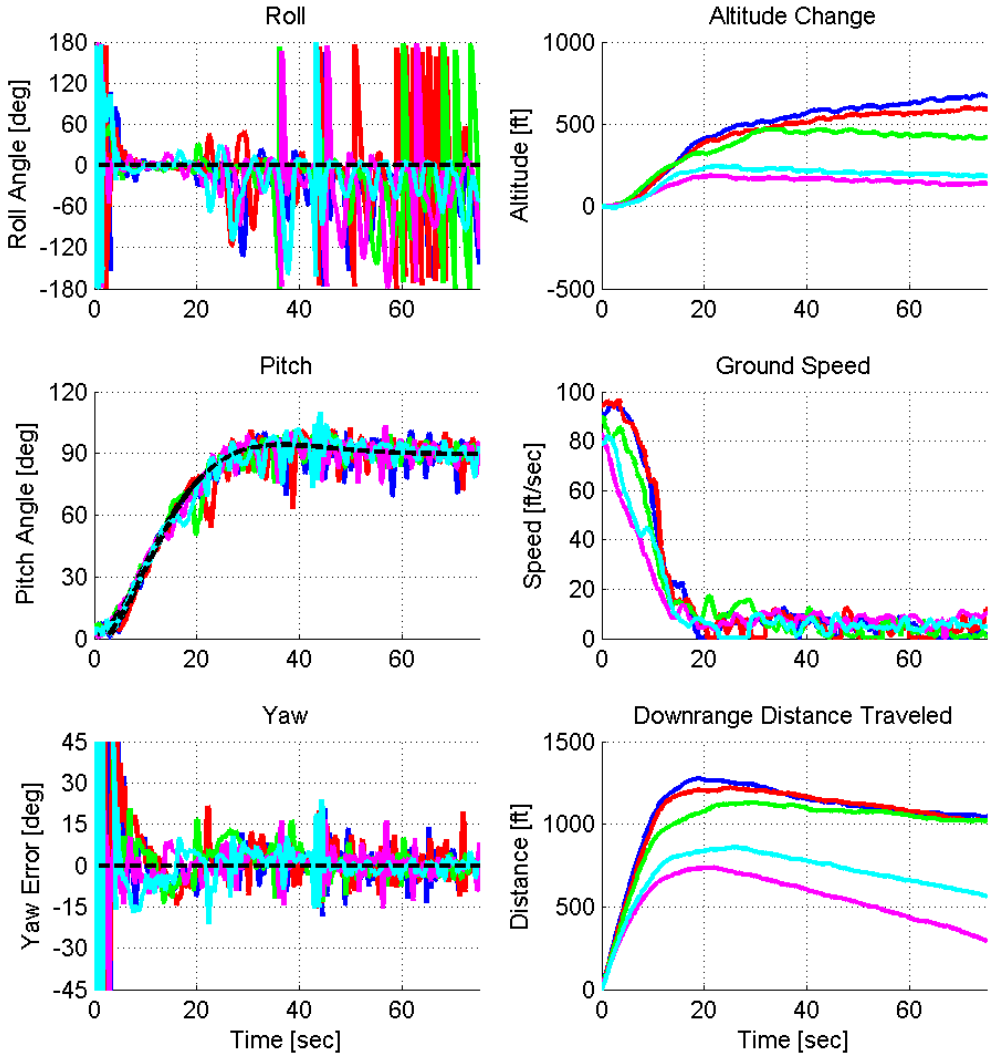
Flight Test: PID Reference Model Response, Rise Time = 15 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	6.9	yes	419.3	974.6
2	1.0	yes	557.2	938.7
3	10.5	yes	230.0	716.6
4	8.2	yes	349.4	845.5
5	6.2	yes	497.3	679.7
	Average		410.6	831.0
	Median		419.3	845.5
	Std. Dev.		127.9	130.8



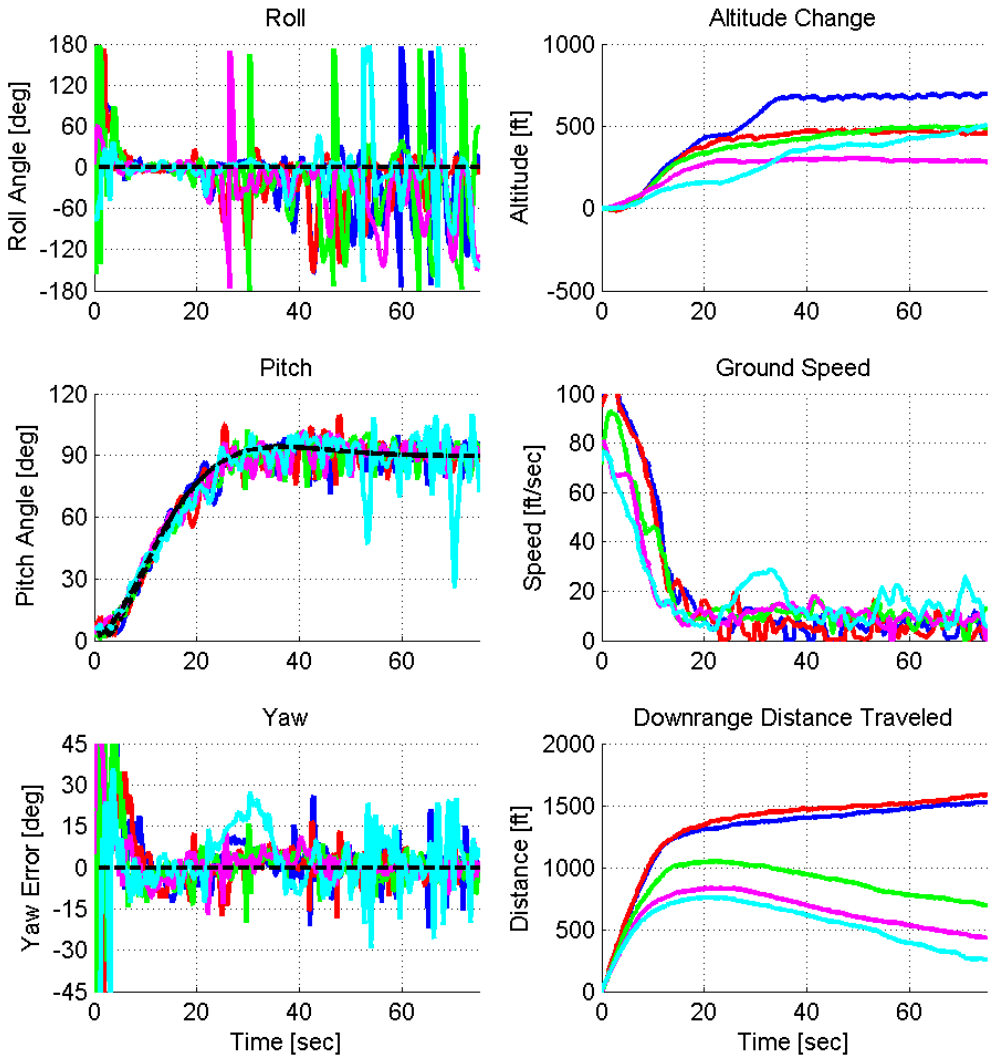
Flight Test: PID Reference Model Response, Rise Time = 15 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	5.6	yes	680.9	1274.6
2	4.9	yes	599.7	1218.9
3	9.8	yes	469.4	1130.9
4	17.1	yes	187.0	736.9
5	8.2	yes	246.9	859.0
	Average		436.8	1044.1
	Median		469.4	1130.9
	Std. Dev.		215.4	234.4



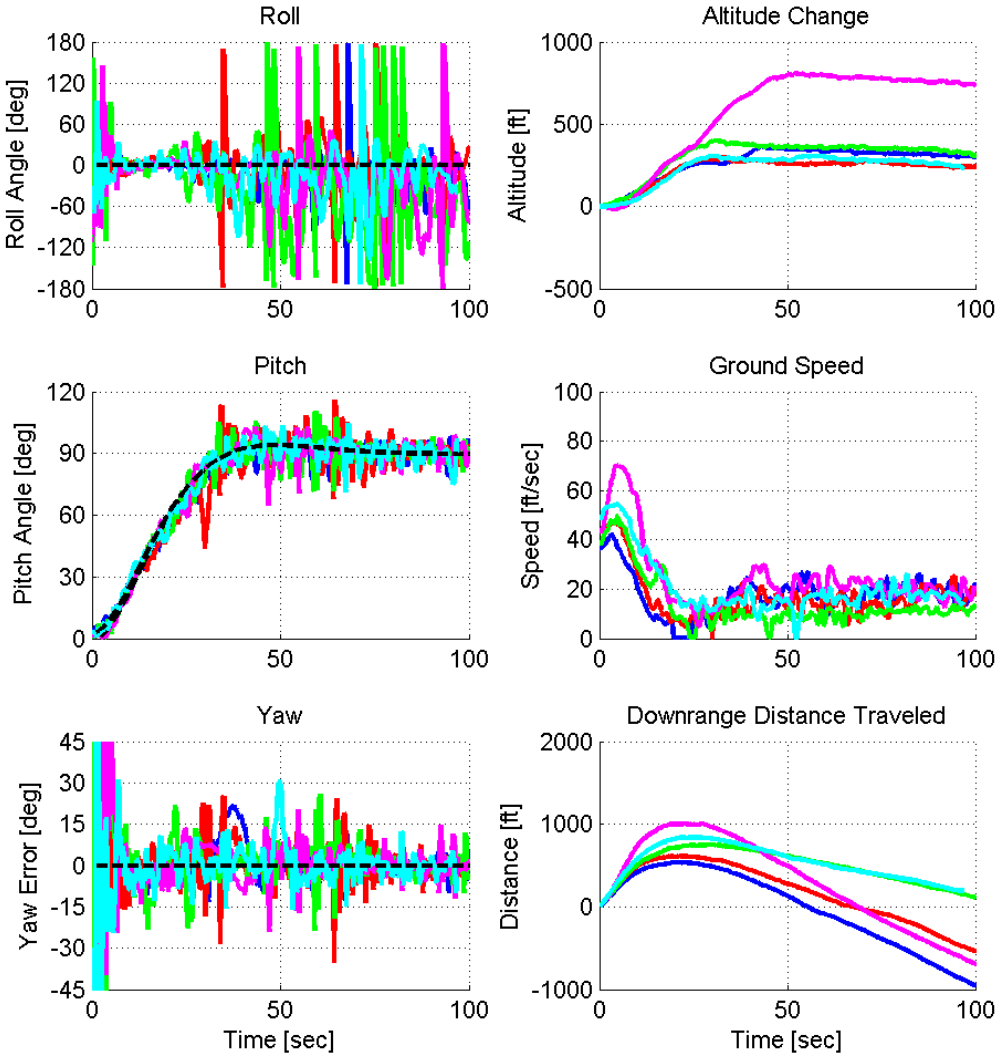
Flight Test: PID Reference Model Response, Rise Time = 15 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	1.0	yes	697.2	1524.0
2	3.3	yes	480.3	1584.1
3	3.9	yes	498.9	1049.7
4	13.8	yes	306.8	833.1
5	16.1	yes	508.3	757.7
	Average		498.3	1149.7
	Median		498.9	1049.7
	Std. Dev.		138.5	384.9



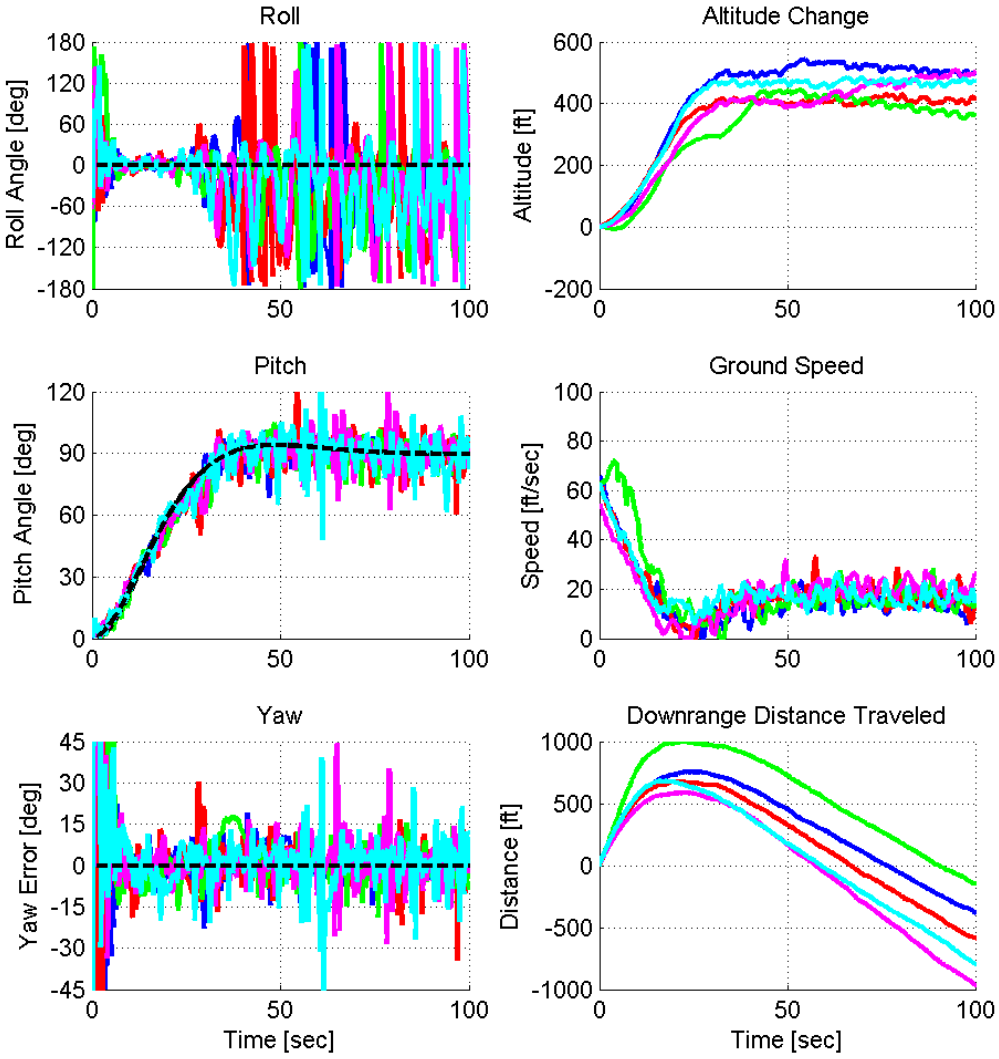
Flight Test: PID Reference Model Response, Rise Time = 20 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	13.1	yes	363.9	533.6
2	10.5	yes	297.4	610.8
3	4.6	yes	398.9	753.6
4	12.8	yes	807.4	1005.3
5	8.9	yes	306.8	841.4
	Average		434.9	748.9
	Median		363.9	753.6
	Std. Dev.		212.4	186.9



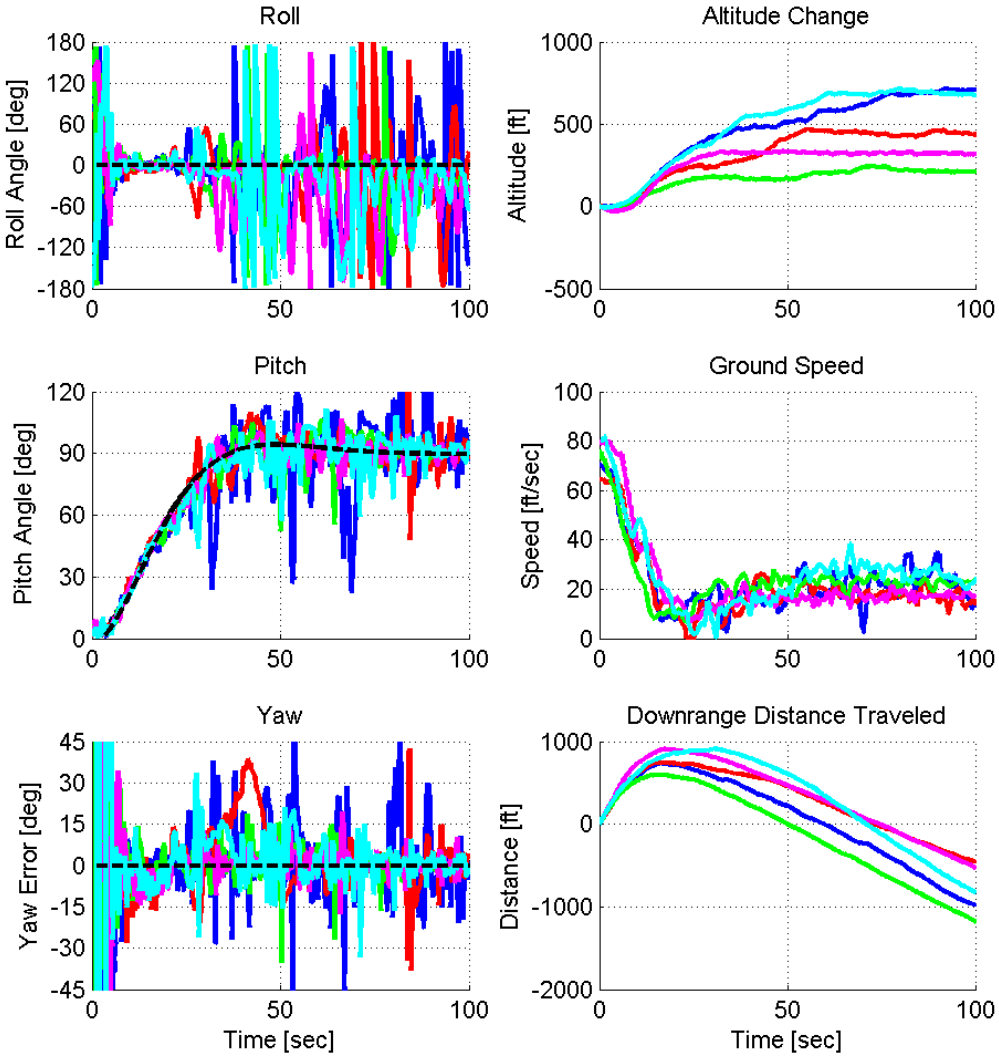
Flight Test: PID Reference Model Response, Rise Time = 20 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	9.5	yes	544.6	754.9
2	9.5	yes	430.2	676.1
3	12.1	yes	443.6	999.6
4	9.8	yes	508.1	588.2
5	6.6	yes	483.8	679.9
	Average		482.0	739.8
	Median		483.8	679.9
	Std. Dev.		46.8	156.8



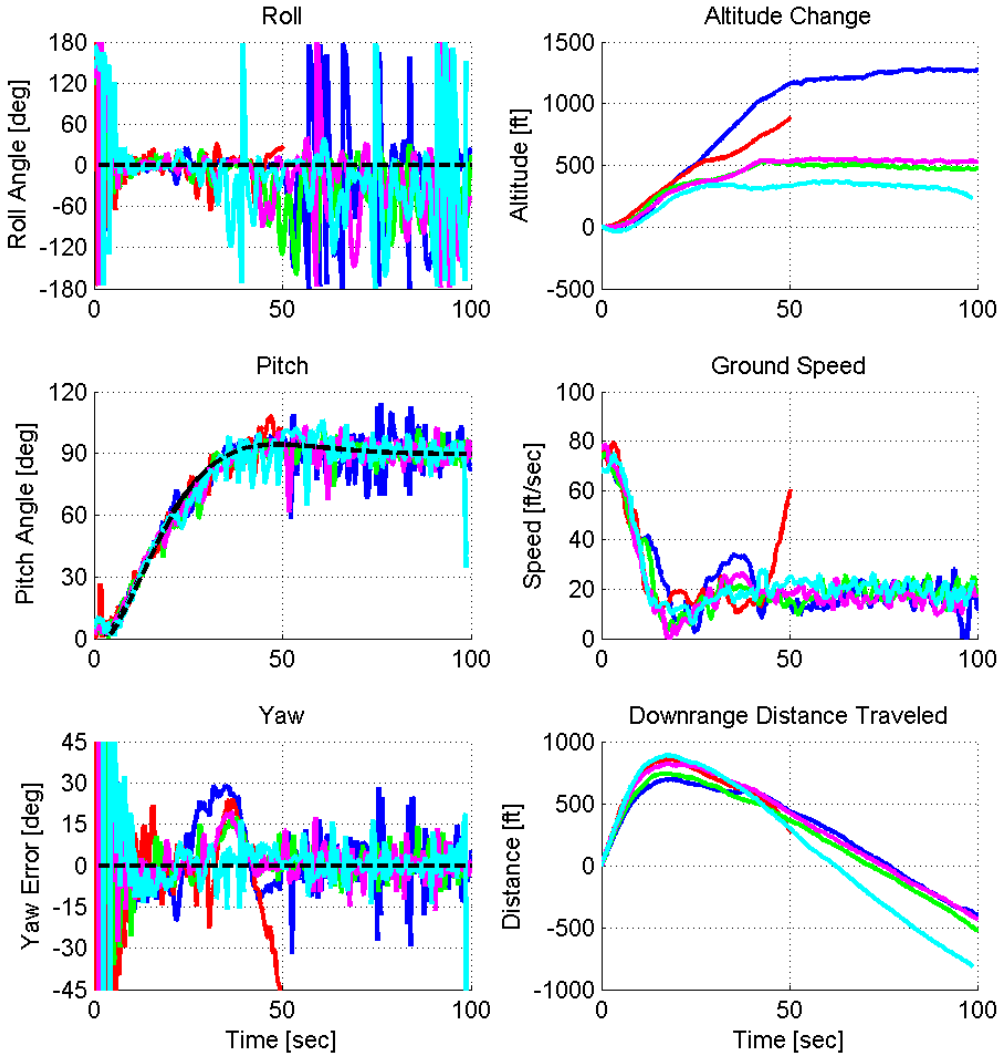
Flight Test: PID Reference Model Response, Rise Time = 20 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	15.7	yes	718.7	733.6
2	15.7	yes	469.9	747.1
3	13.8	yes	244.9	596.6
4	15.4	yes	335.1	908.8
5	9.5	yes	718.1	909.7
	Average		497.4	779.2
	Median		469.9	747.1
	Std. Dev.		217.1	132.5



Flight Test: PID Reference Model Response, Rise Time = 20 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	9.8	yes	1280.9	694.9
2	10.5	no	883.8	856.0
3	9.8	yes	530.5	741.1
4	15.4	yes	554.0	816.1
5	13.1	yes	372.0	886.5
	Average		684.4	784.7
	Median		542.3	778.6
	Std. Dev.		405.8	84.3



B.3 Model Reference Adaptive Control

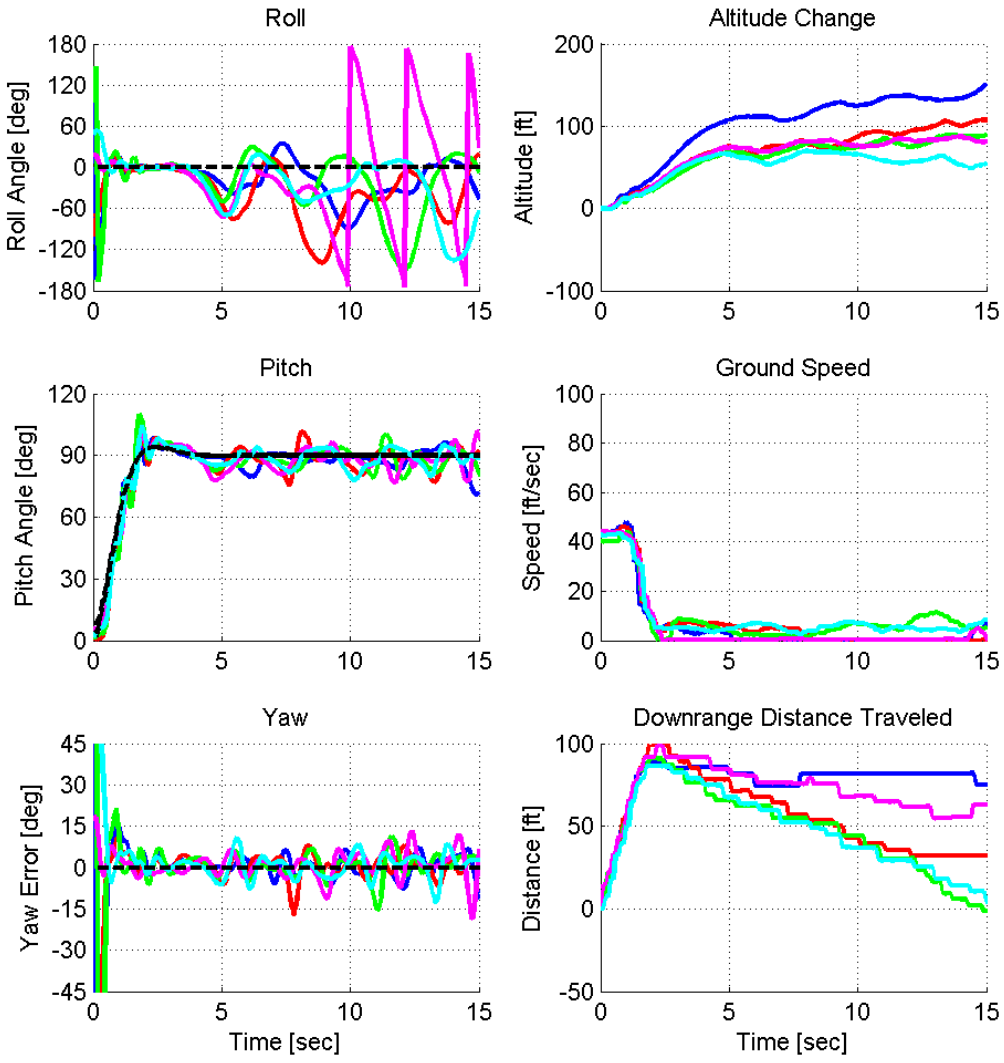
MRAC Flight Tests: Probability of Success					
Rise Time [sec]	Approach Speed [ft/sec]				Cumulative
	40	60	80	100	
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
5	100%	80%	60%	100%	85%
7	100%	100%	80%	80%	90%
10	100%	60%	20%	40%	55%
15	40%	40%	20%	40%	35%
20	0%	0%	0%	0%	0%
Cumulative	80%	73%	60%	70%	71%

MRAC Flight Tests				
Rise Time	Approach Speed	Max. Altitude Change [ft]		
		Average	Median	Std. Dev.
1 sec	40 ft/sec	101.0	89.2	30.9
	60 ft/sec	94.3	94.8	10.1
	80 ft/sec	105.6	100.0	31.8
	100 ft/sec	110.5	99.9	21.1
2 sec	40 ft/sec	120.7	108.5	43.2
	60 ft/sec	157.1	144.9	43.6
	80 ft/sec	139.5	138.0	20.7
	100 ft/sec	147.7	142.5	15.6
3 sec	40 ft/sec	76.1	91.1	35.7
	60 ft/sec	113.1	111.6	55.8
	80 ft/sec	106.4	109.4	15.8
	100 ft/sec	192.7	178.4	31.7
5 sec	40 ft/sec	70.4	68.0	19.9
	60 ft/sec	107.3	101.7	37.3
	80 ft/sec	119.0	119.9	20.0
	100 ft/sec	142.7	140.0	30.7
7 sec	40 ft/sec	149.3	149.0	46.6
	60 ft/sec	140.0	145.5	47.9
	80 ft/sec	141.4	143.5	29.8
	100 ft/sec	165.2	161.3	20.0
10 sec	40 ft/sec	100.4	75.7	43.2
	60 ft/sec	78.4	79.7	8.6
	80 ft/sec	133.7	133.7	N/A
	100 ft/sec	149.1	149.1	38.0
15 sec	40 ft/sec	142.9	142.9	2.4
	60 ft/sec	268.7	268.7	78.8
	80 ft/sec	221.3	221.3	N/A
	100 ft/sec	252.9	252.9	42.2
20 sec	40 ft/sec	N/A	N/A	N/A
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	N/A	N/A	N/A

MRAC Flight Tests				
Rise Time	Approach Speed	Max. Distance Traveled [ft]		
		Average	Median	Std. Dev.
1 sec	40 ft/sec	92.9	90.9	5.9
	60 ft/sec	126.5	130.3	12.9
	80 ft/sec	159.1	155.9	13.6
	100 ft/sec	149.5	145.4	13.3
2 sec	40 ft/sec	107.4	106.7	8.4
	60 ft/sec	136.3	141.1	11.7
	80 ft/sec	124.8	125.9	24.9
	100 ft/sec	143.3	146.7	17.8
3 sec	40 ft/sec	180.2	164.2	28.6
	60 ft/sec	214.8	215.6	6.8
	80 ft/sec	278.2	272.8	22.4
	100 ft/sec	317.0	309.6	13.0
5 sec	40 ft/sec	306.4	297.5	32.3
	60 ft/sec	447.5	472.9	126.4
	80 ft/sec	497.5	515.2	90.4
	100 ft/sec	366.0	373.6	35.8
7 sec	40 ft/sec	351.2	338.6	25.2
	60 ft/sec	425.8	421.3	44.3
	80 ft/sec	449.2	431.1	48.0
	100 ft/sec	613.9	590.0	90.9
10 sec	40 ft/sec	503.5	462.8	130.3
	60 ft/sec	599.0	576.6	70.2
	80 ft/sec	612.8	612.8	N/A
	100 ft/sec	654.8	654.8	20.0
15 sec	40 ft/sec	699.4	699.4	71.2
	60 ft/sec	877.0	877.0	28.9
	80 ft/sec	960.1	960.1	N/A
	100 ft/sec	1088.4	1088.4	99.2
20 sec	40 ft/sec	N/A	N/A	N/A
	60 ft/sec	N/A	N/A	N/A
	80 ft/sec	N/A	N/A	N/A
	100 ft/sec	N/A	N/A	N/A

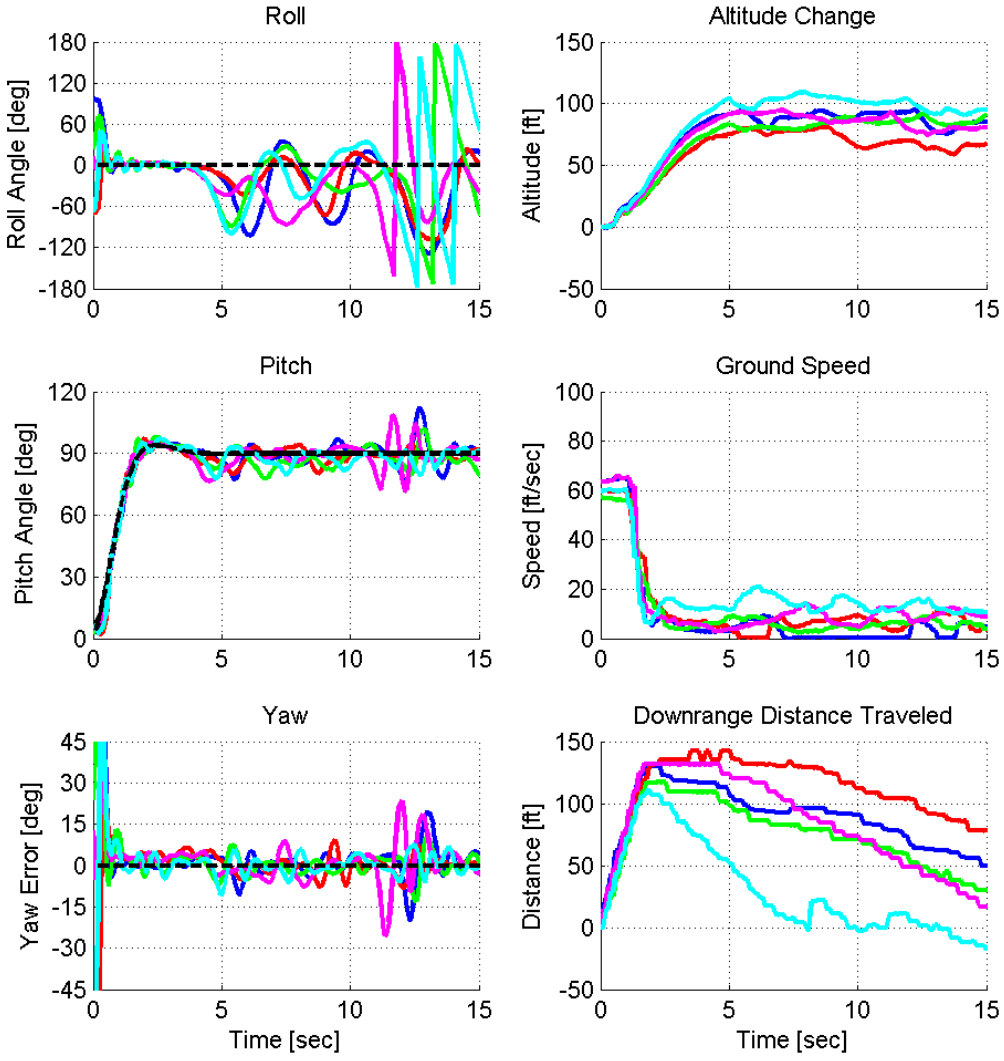
Flight Test: MRAC, Rise Time = 1 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	2.3	yes	150.8	88.8
2	4.9	yes	107.8	99.5
3	4.6	yes	89.2	90.9
4	7.5	yes	87.6	98.8
5	2.6	yes	69.7	86.4
	Average		101.0	92.9
	Median		89.2	90.9
	Std. Dev.		30.9	5.9



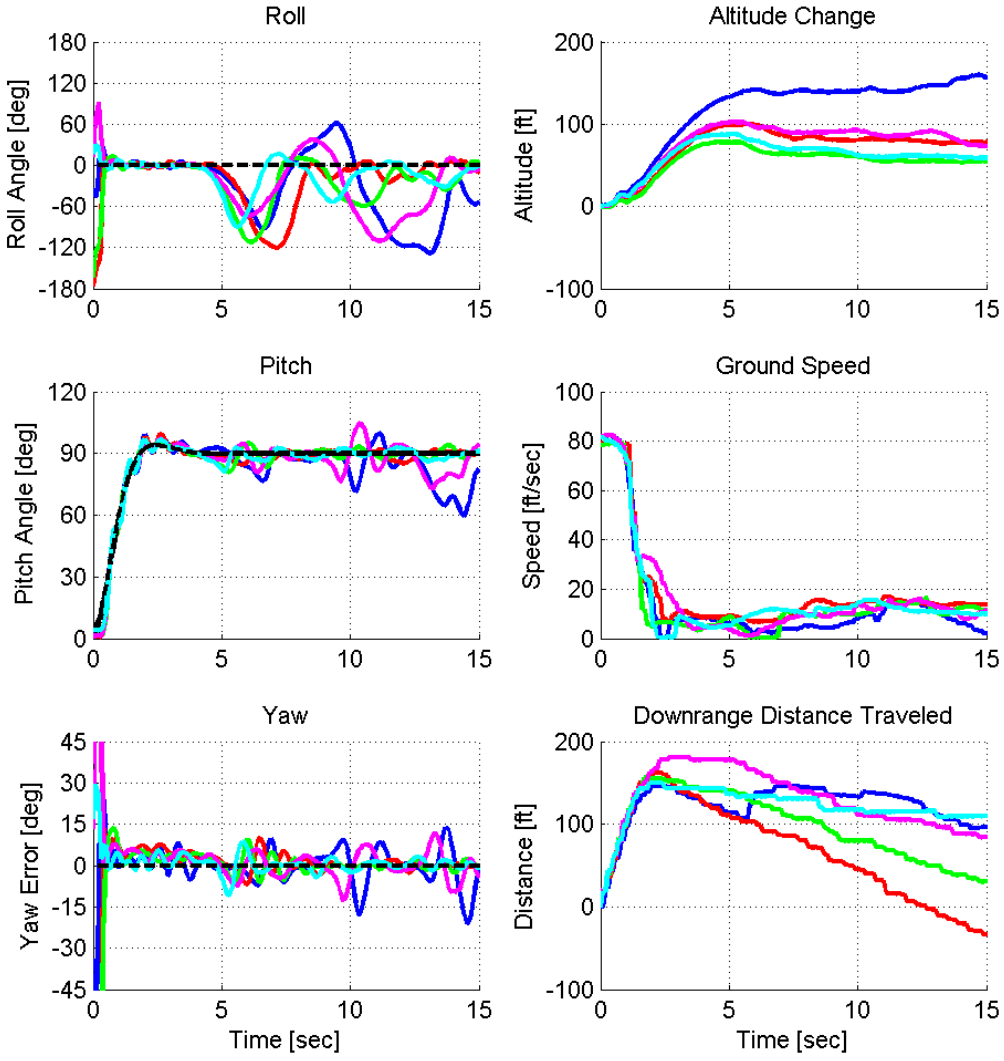
Flight Test: MRAC, Rise Time = 1 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	95.1	130.3
2	8.5	yes	80.8	142.9
3	3.9	yes	91.7	117.3
4	4.9	yes	94.8	132.0
5	4.3	yes	109.1	110.1
	Average		94.3	126.5
	Median		94.8	130.3
	Std. Dev.		10.1	12.9



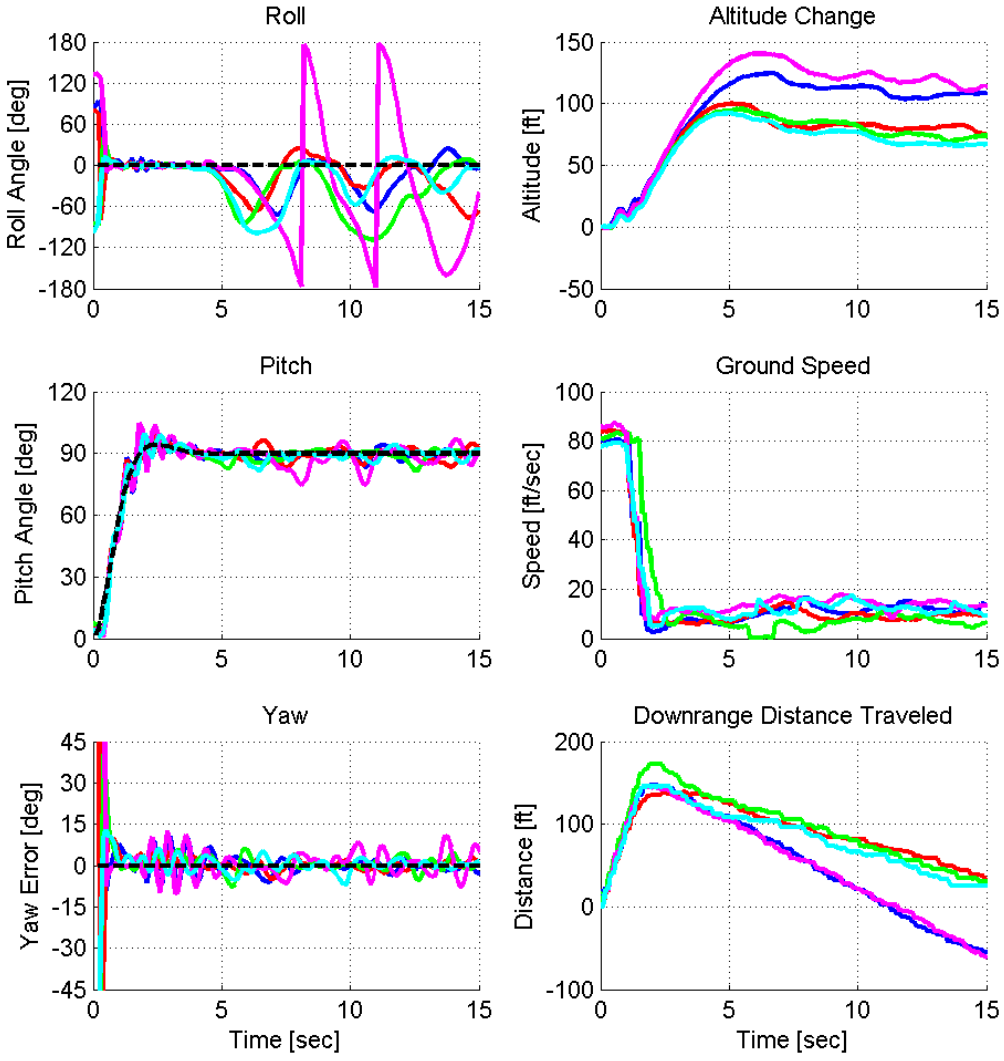
Flight Test: MRAC, Rise Time = 1 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	159.7	146.3
2	0.7	yes	100.0	161.7
3	8.5	yes	78.0	155.9
4	1.0	yes	102.5	181.1
5	0.0	yes	87.9	150.4
	Average		105.6	159.1
	Median		100.0	155.9
	Std. Dev.		31.8	13.6



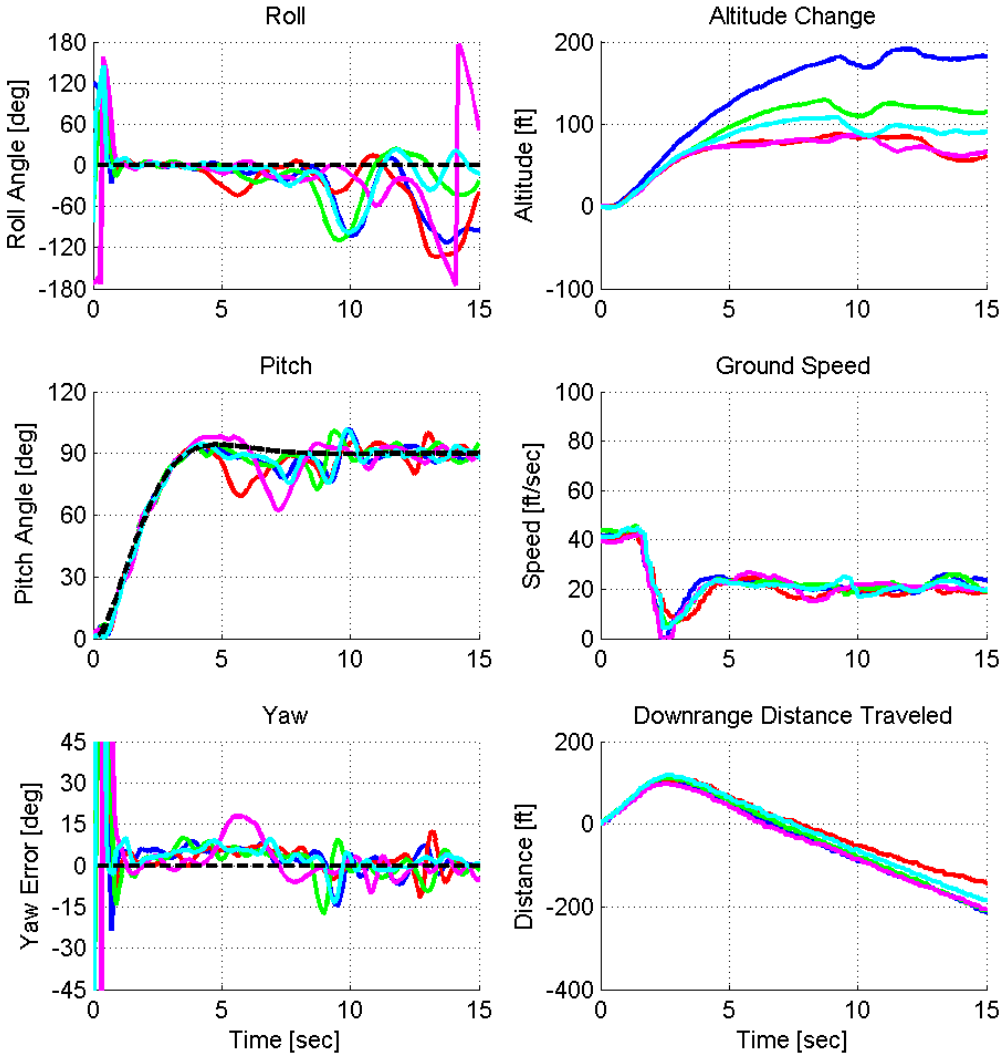
Flight Test: MRAC, Rise Time = 1 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	2.6	yes	124.5	146.5
2	7.5	yes	99.9	138.6
3	7.9	yes	95.7	172.7
4	9.8	yes	140.6	144.4
5	2.6	yes	91.8	145.4
	Average		110.5	149.5
	Median		99.9	145.4
	Std. Dev.		21.1	13.3



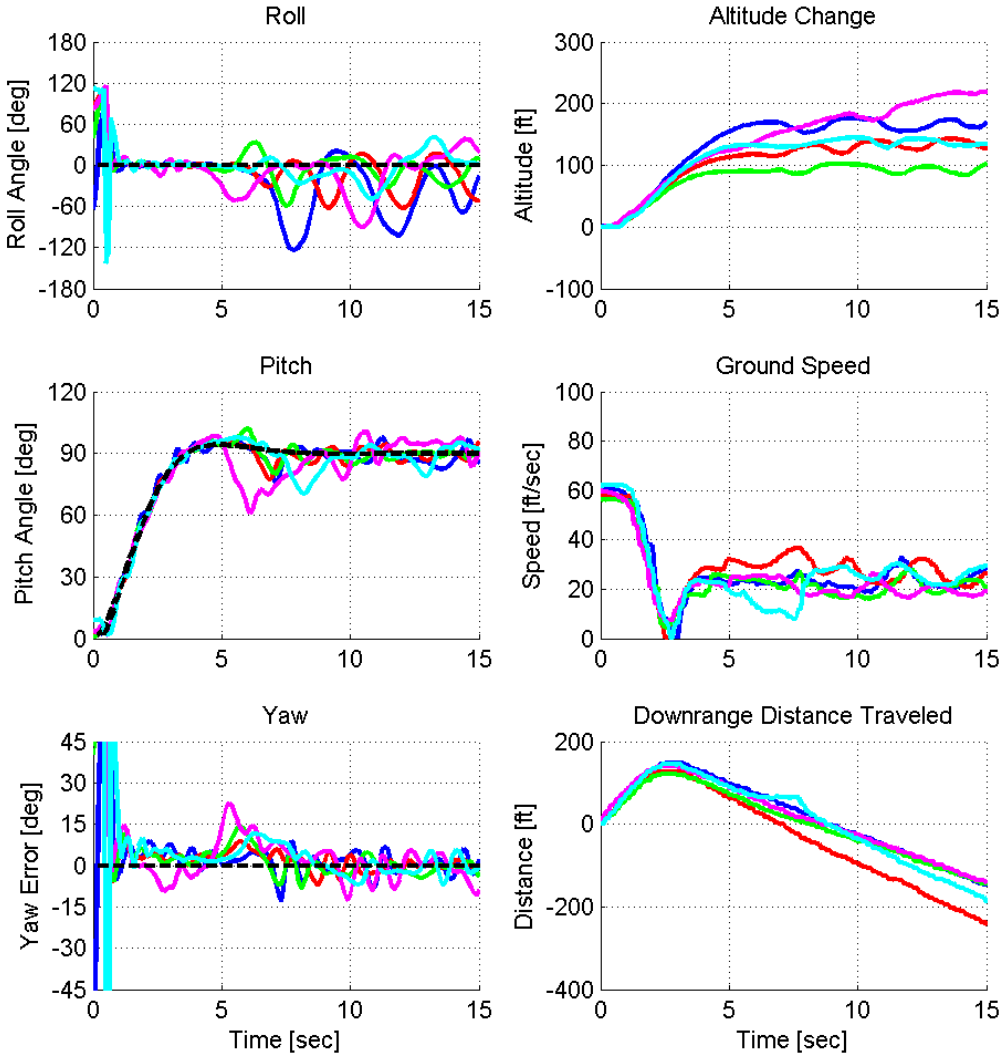
Flight Test: MRAC, Rise Time = 2 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	7.5	yes	191.2	105.2
2	7.2	yes	87.7	106.7
3	6.2	yes	129.5	110.7
4	6.9	yes	86.6	95.5
5	6.2	yes	108.5	118.7
	Average		120.7	107.4
	Median		108.5	106.7
	Std. Dev.		43.2	8.4



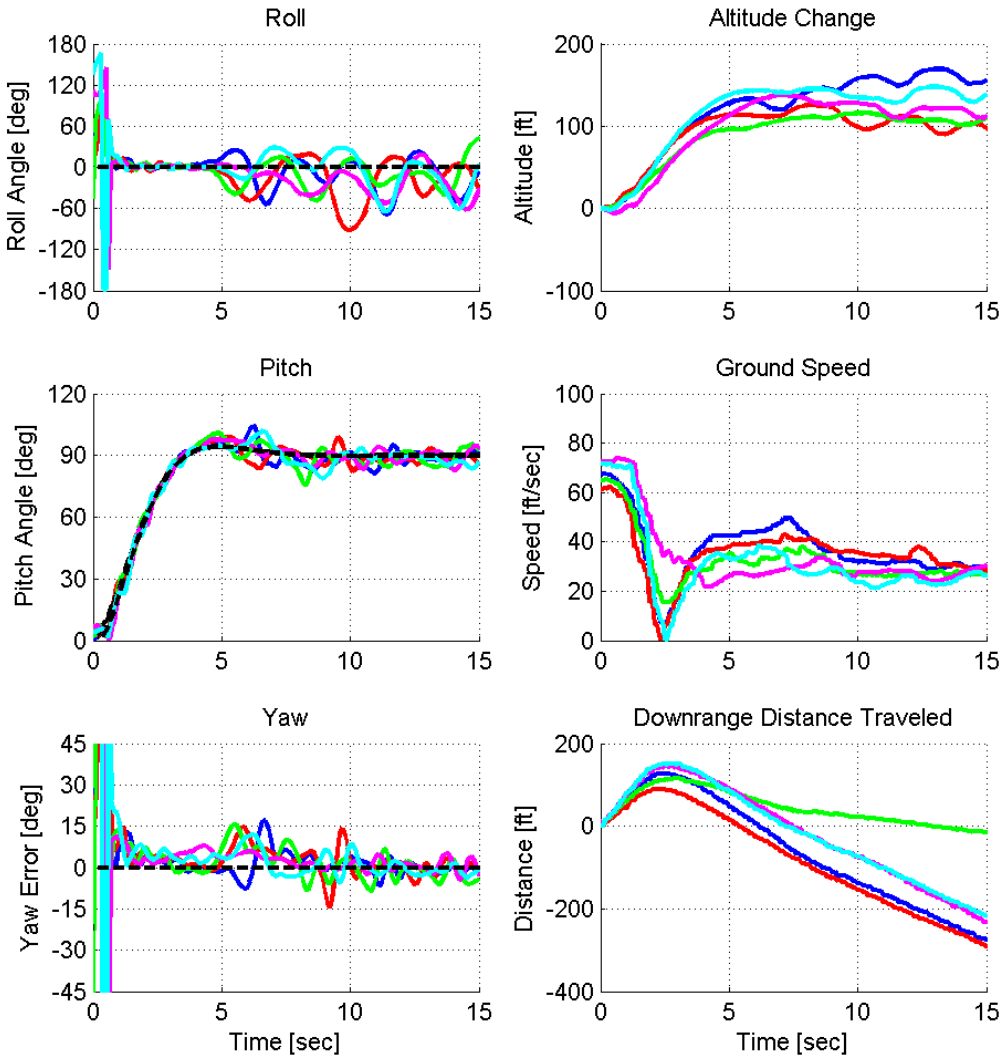
Flight Test: MRAC, Rise Time = 2 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	9.8	yes	175.9	147.7
2	11.8	yes	142.7	127.1
3	7.2	yes	102.5	120.9
4	2.6	yes	219.7	141.1
5	8.9	yes	144.9	144.8
	Average		157.1	136.3
	Median		144.9	141.1
	Std. Dev.		43.6	11.7



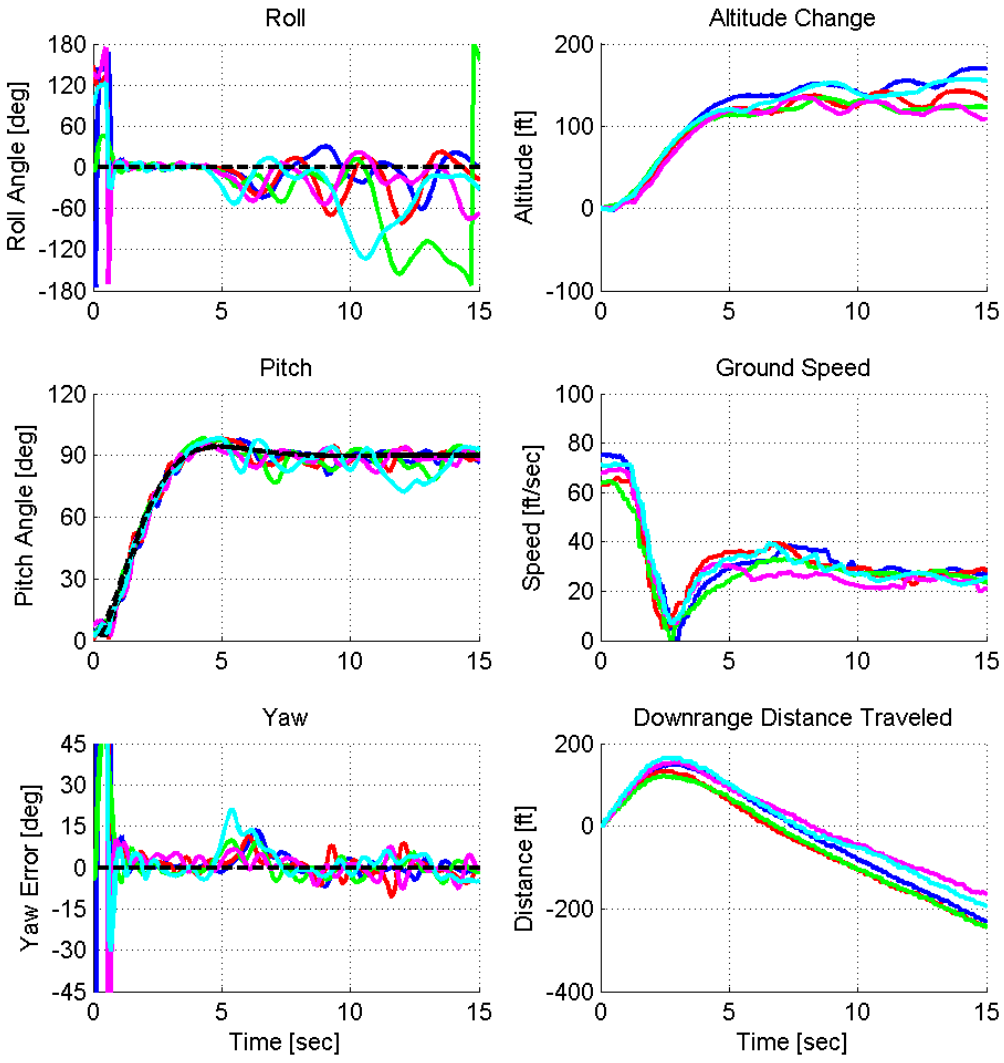
Flight Test: MRAC, Rise Time = 2 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	15.7	yes	169.6	125.9
2	7.2	yes	125.7	88.2
3	11.2	yes	116.2	114.7
4	5.9	yes	138.0	144.9
5	11.8	yes	147.9	150.0
	Average		139.5	124.8
	Median		138.0	125.9
	Std. Dev.		20.7	24.9



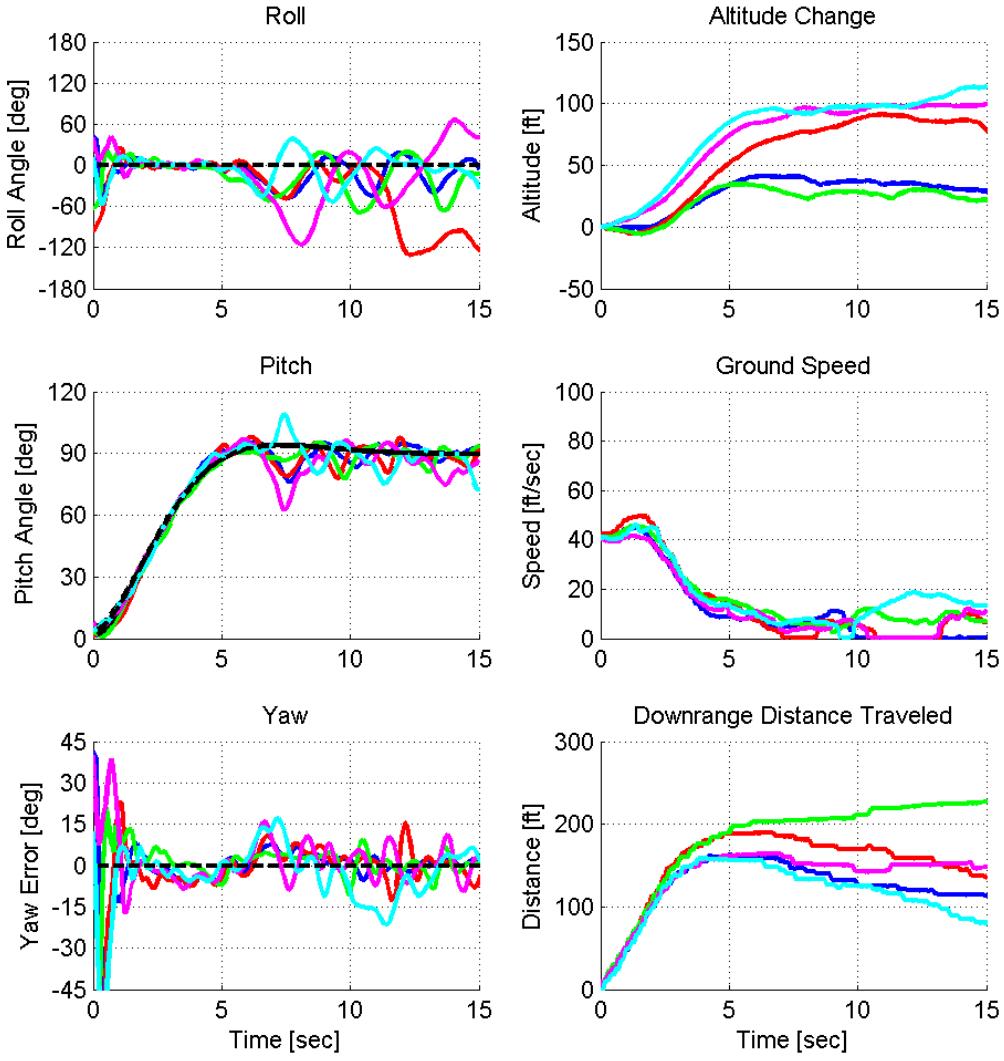
Flight Test: MRAC, Rise Time = 2 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	8.2	yes	170.4	146.7
2	6.2	yes	142.5	131.7
3	7.2	yes	134.3	119.9
4	8.2	yes	134.6	153.0
5	4.6	yes	156.7	165.1
	Average		147.7	143.3
	Median		142.5	146.7
	Std. Dev.		15.6	17.8



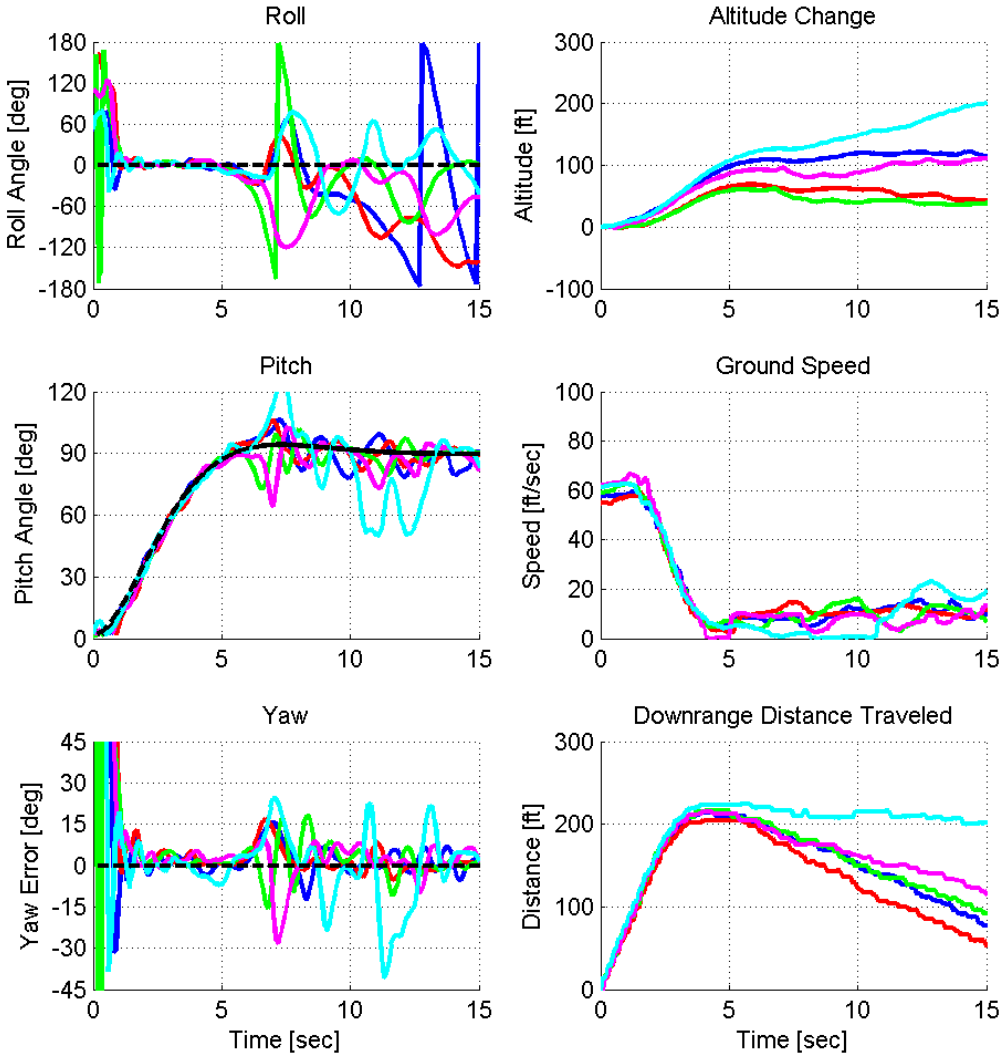
Flight Test: MRAC, Rise Time = 3 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	2.0	yes	41.4	162.9
2	2.3	yes	91.1	189.0
3	0.0	yes	34.7	226.6
4	6.2	yes	99.4	164.2
5	2.0	yes	113.7	158.3
	Average		76.1	180.2
	Median		91.1	164.2
	Std. Dev.		35.7	28.6



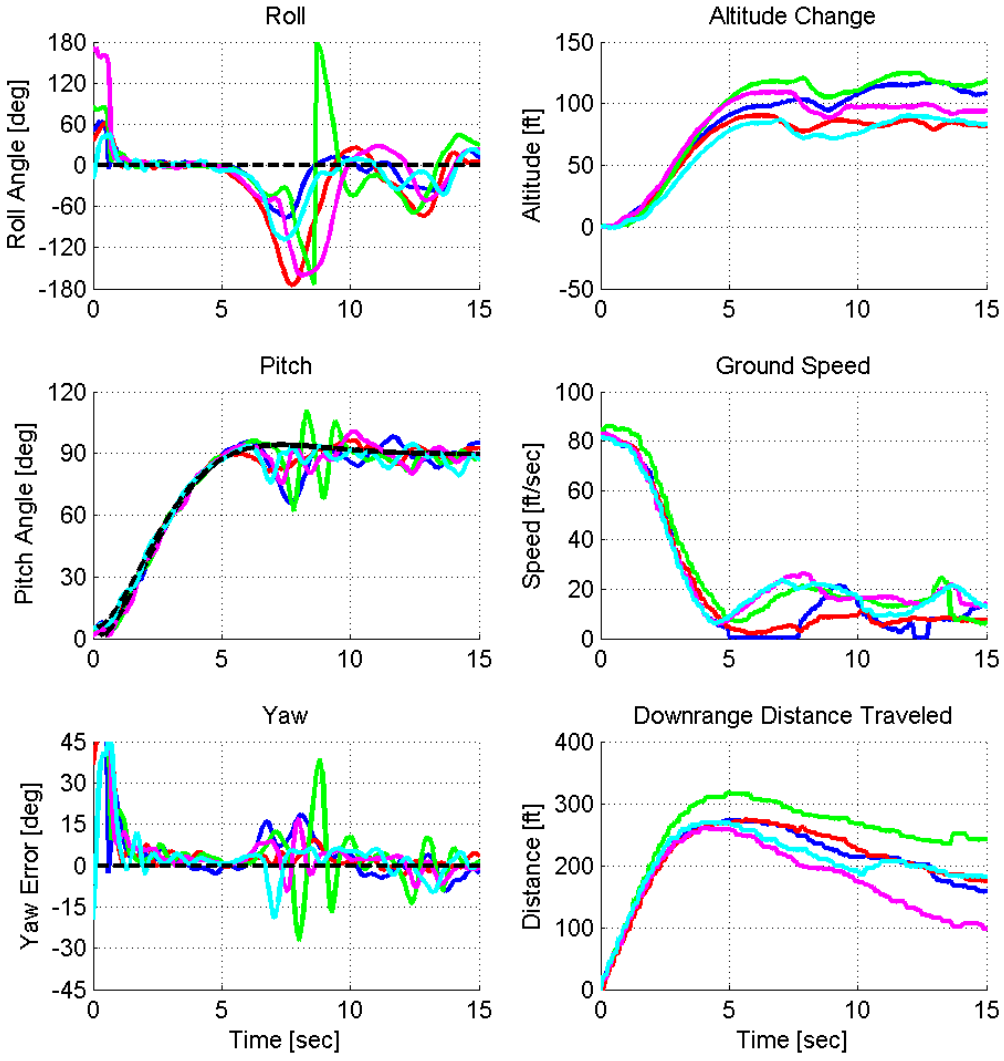
Flight Test: MRAC, Rise Time = 3 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	3.0	yes	122.2	214.0
2	3.0	yes	68.9	204.8
3	7.9	yes	61.7	215.9
4	2.3	yes	111.6	215.6
5	0.0	yes	201.2	223.8
	Average		113.1	214.8
	Median		111.6	215.6
	Std. Dev.		55.8	6.8



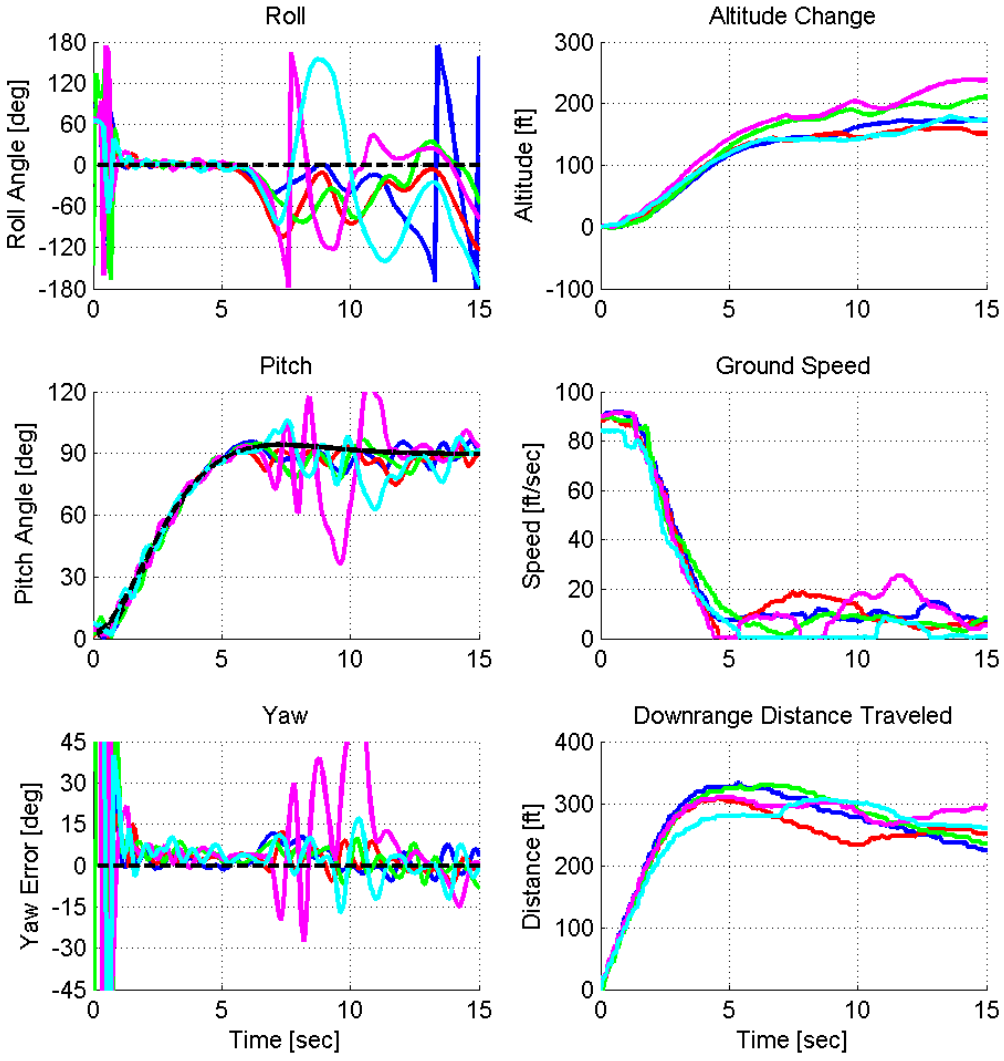
Flight Test: MRAC, Rise Time = 3 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	3.9	yes	117.6	272.8
2	6.2	yes	90.3	273.4
3	6.9	yes	124.7	317.0
4	11.2	yes	109.4	259.3
5	6.9	yes	90.1	268.6
	Average		106.4	278.2
	Median		109.4	272.8
	Std. Dev.		15.8	22.4



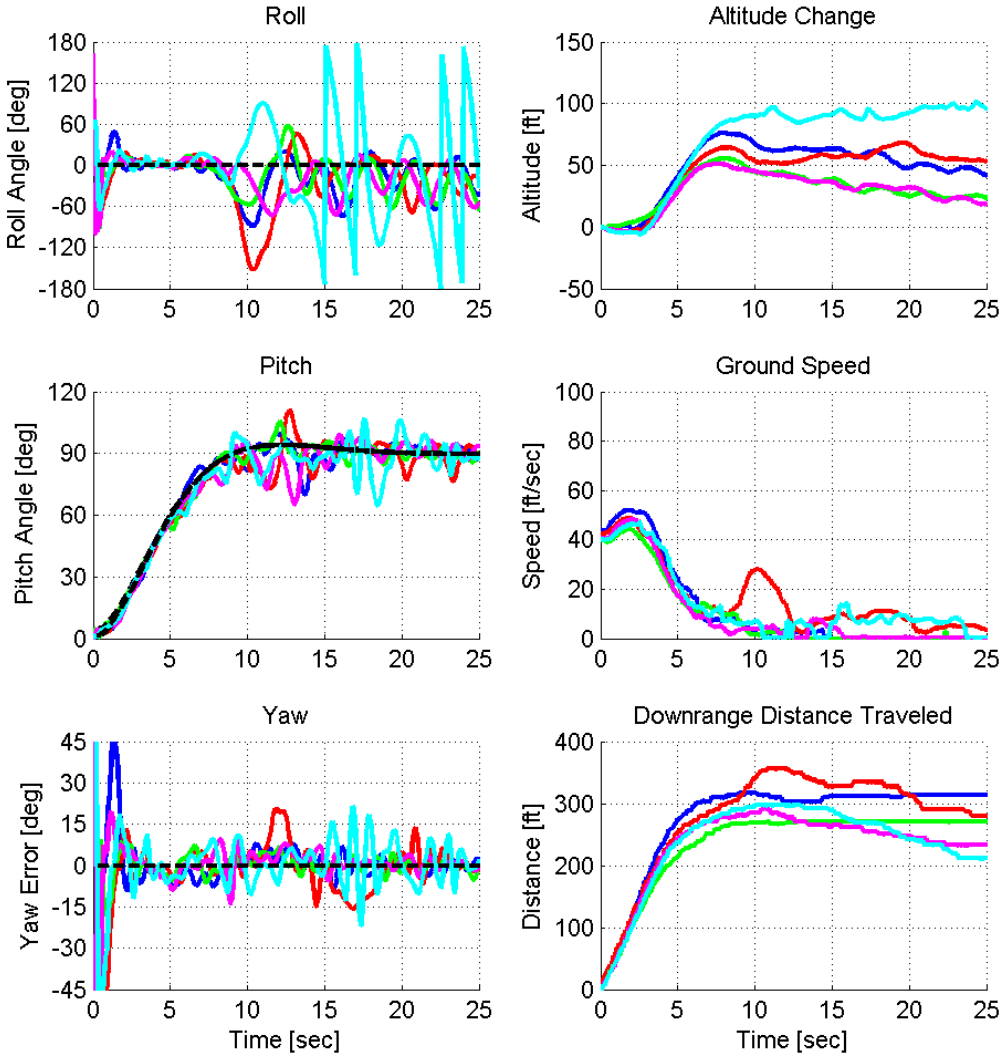
Flight Test: MRAC, Rise Time = 3 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	11.8	yes	175.3	332.6
2	9.8	yes	160.3	307.5
3	4.6	yes	210.4	329.5
4	5.2	yes	239.0	309.6
5	3.0	yes	178.4	305.6
	Average		192.7	317.0
	Median		178.4	309.6
	Std. Dev.		31.7	13.0



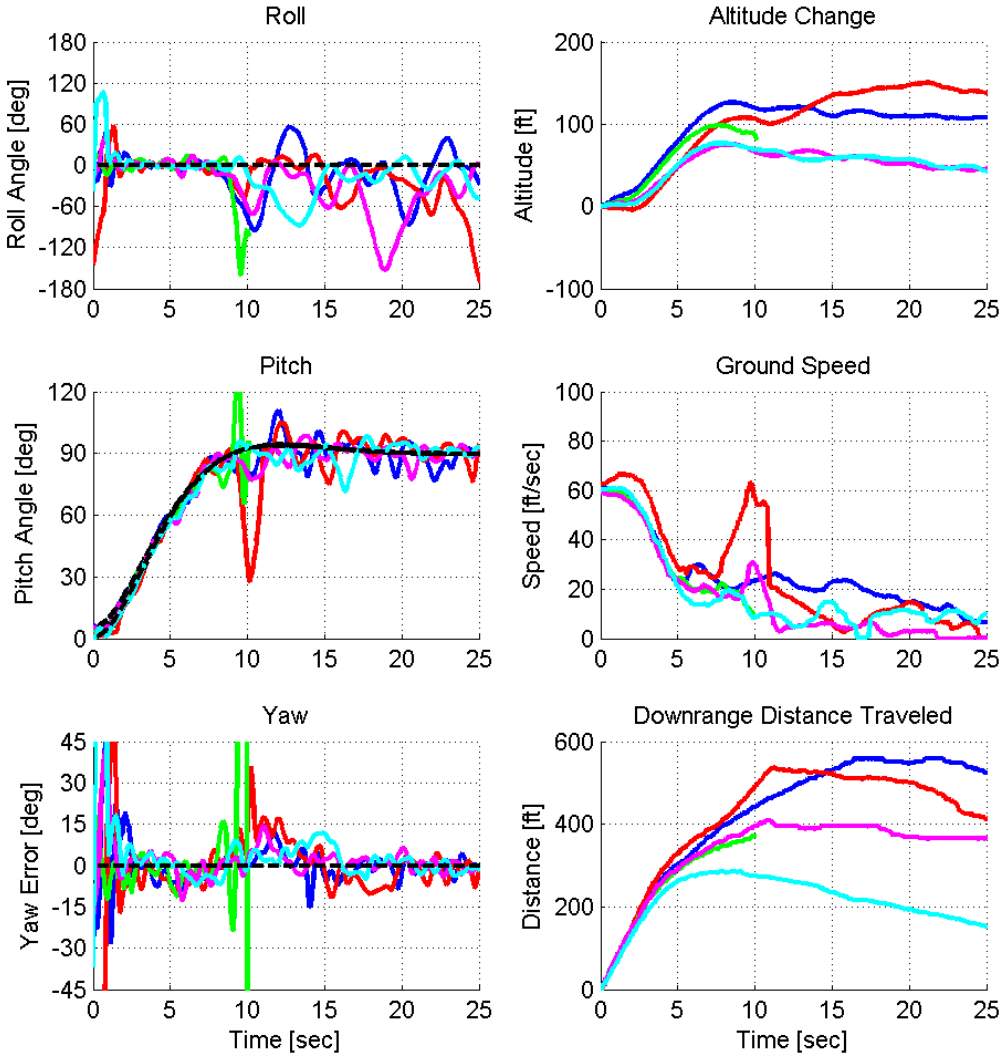
Flight Test: MRAC, Rise Time = 5 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	6.2	yes	76.2	316.4
2	4.9	yes	68.0	356.0
3	2.3	yes	55.4	270.1
4	7.5	yes	51.1	291.9
5	4.9	yes	101.1	297.5
	Average		70.4	306.4
	Median		68.0	297.5
	Std. Dev.		19.9	32.3



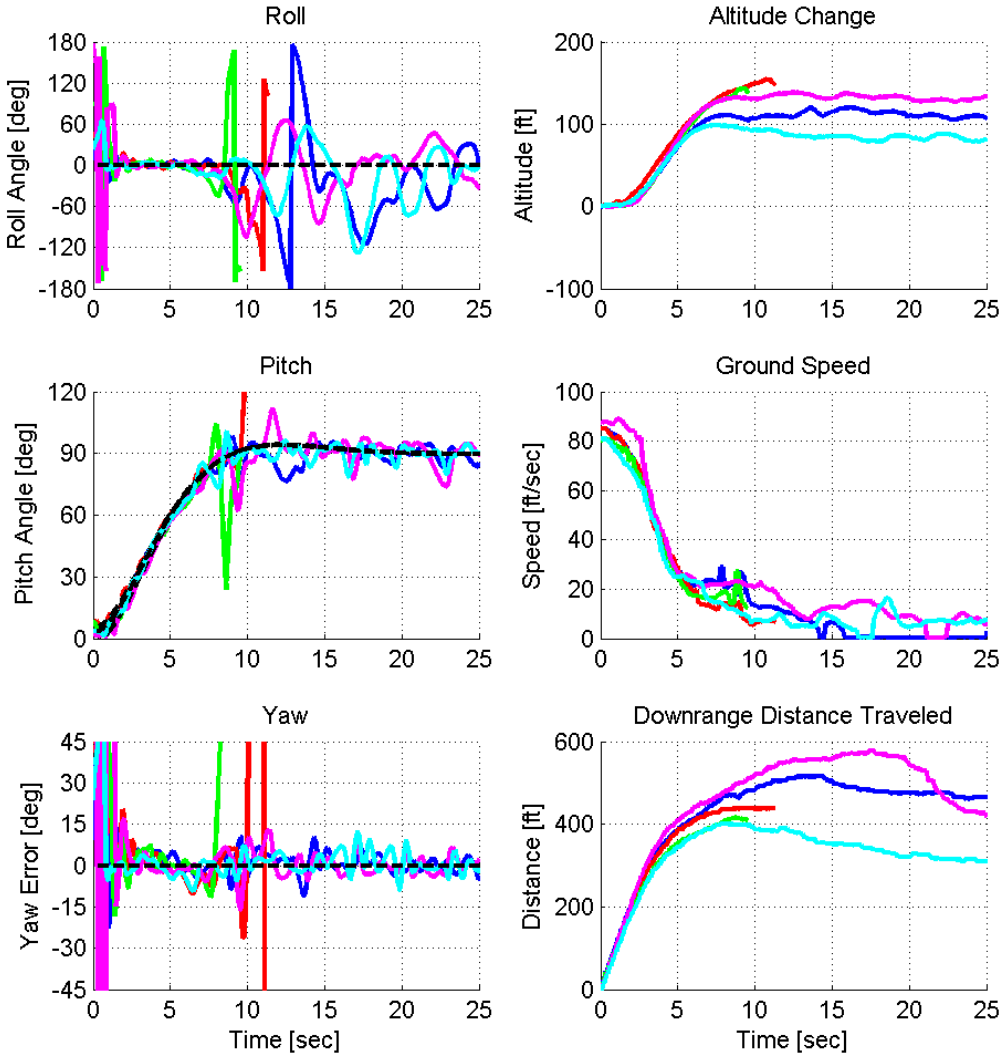
Flight Test: MRAC, Rise Time = 5 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	126.5	558.7
2	0.0	yes	150.4	536.2
3	0.0	no	98.5	370.2
4	0.0	yes	75.4	409.7
5	4.9	yes	76.8	285.5
	Average		107.3	447.5
	Median		101.7	472.9
	Std. Dev.		37.3	126.4



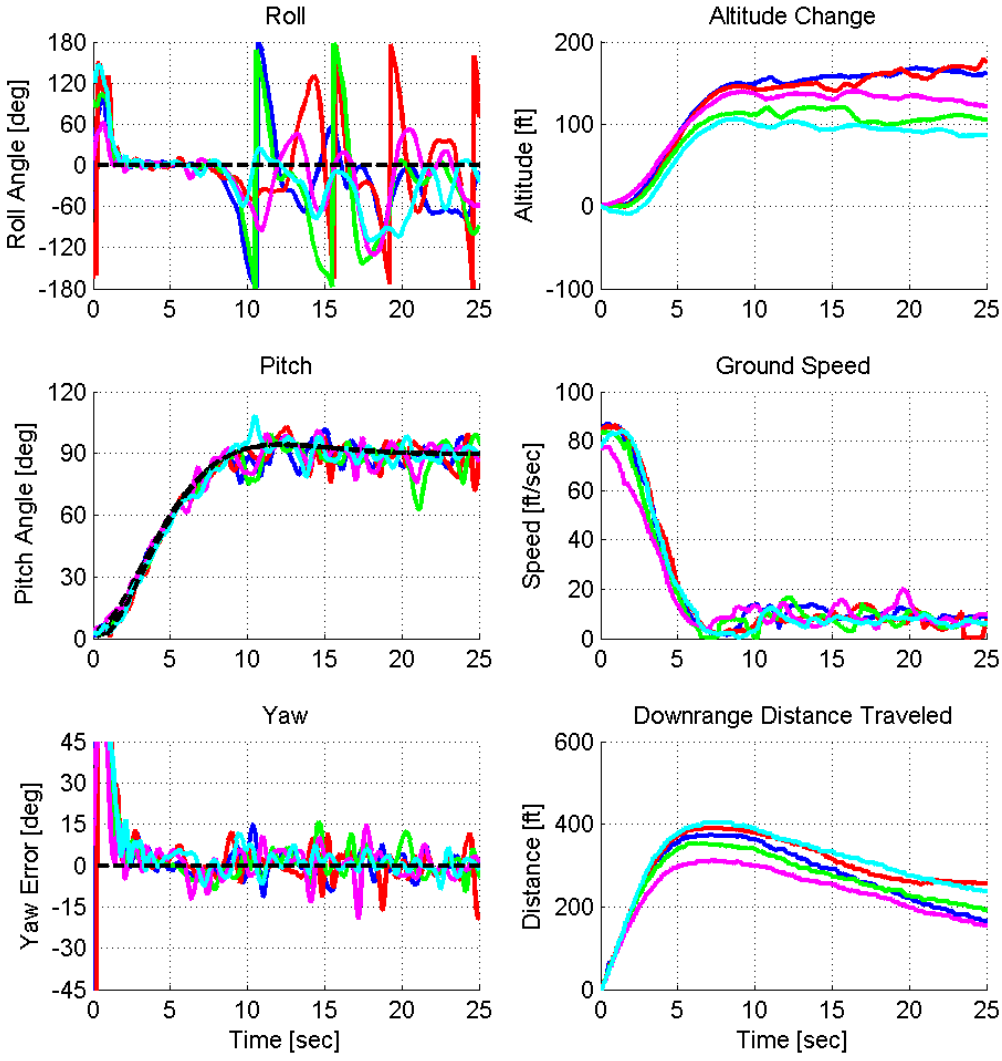
Flight Test: MRAC, Rise Time = 5 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	119.9	515.2
2	2.3	no	154.5	438.2
3	0.0	no	143.9	415.3
4	1.3	yes	138.6	577.8
5	6.2	yes	98.6	399.6
	Average		119.0	497.5
	Median		119.9	515.2
	Std. Dev.		20.0	90.4



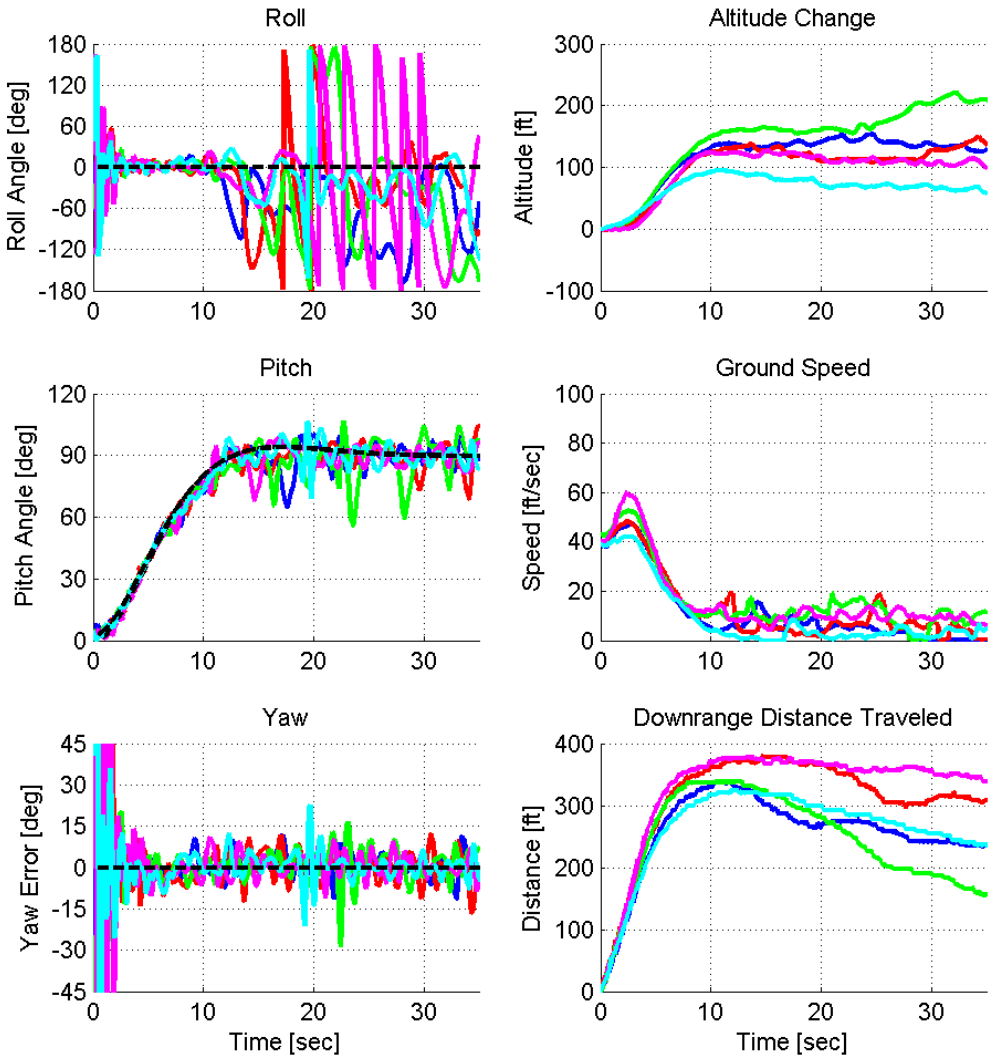
Flight Test: MRAC, Rise Time = 5 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	167.8	373.6
2	3.3	yes	178.9	388.6
3	4.3	yes	120.5	352.2
4	4.9	yes	140.0	311.9
5	1.0	yes	106.1	404.0
	Average		142.7	366.0
	Median		140.0	373.6
	Std. Dev.		30.7	35.8



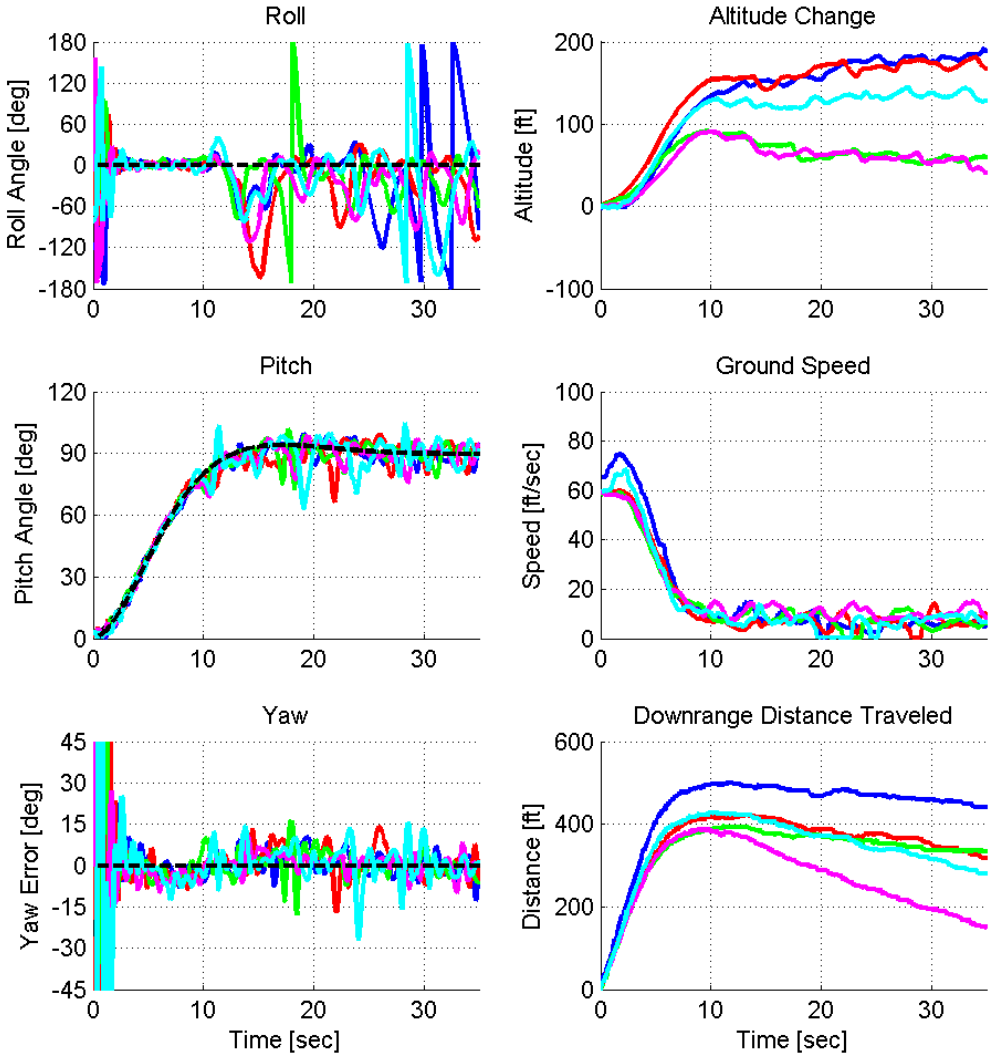
Flight Test: MRAC, Rise Time = 7 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	3.3	yes	153.7	337.1
2	10.2	yes	149.0	378.8
3	4.9	yes	221.6	338.6
4	0.0	yes	127.1	377.5
5	13.1	yes	95.2	324.2
	Average		149.3	351.2
	Median		149.0	338.6
	Std. Dev.		46.6	25.2



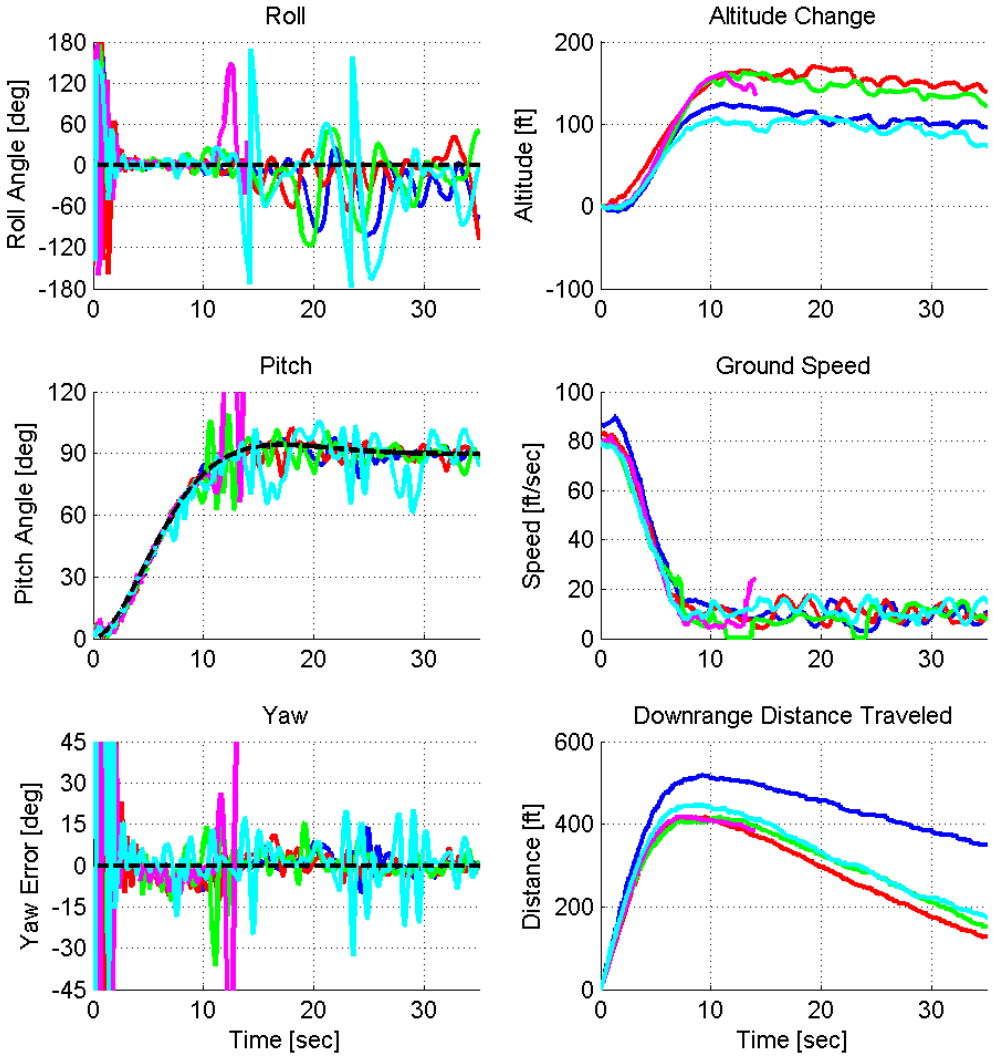
Flight Test: MRAC, Rise Time = 7 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	3.3	yes	191.4	498.8
2	5.6	yes	181.2	421.3
3	9.5	yes	91.3	394.0
4	5.6	yes	90.6	387.1
5	6.9	yes	145.5	427.9
	Average		140.0	425.8
	Median		145.5	421.3
	Std. Dev.		47.9	44.3



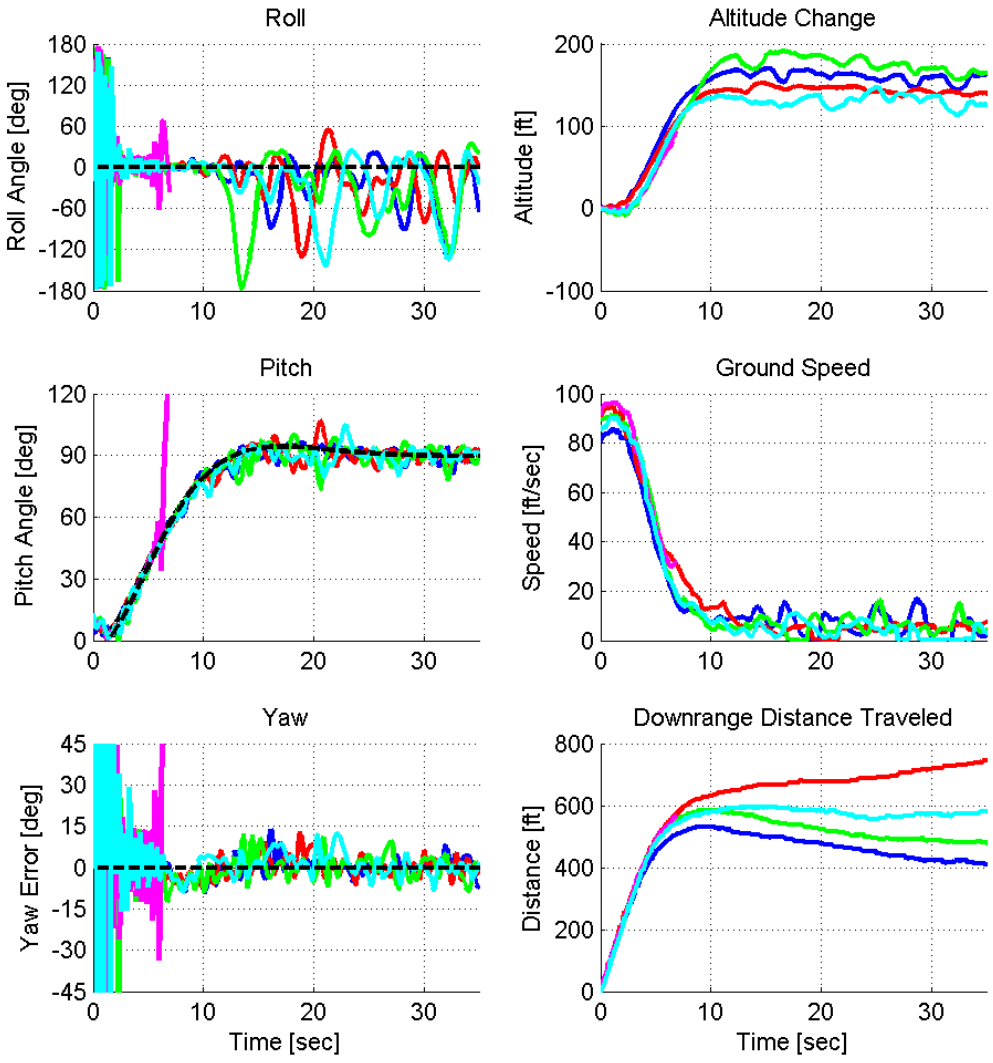
Flight Test: MRAC, Rise Time = 7 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	5.6	yes	124.4	518.2
2	1.0	yes	170.2	416.6
3	0.7	yes	162.6	416.3
4	3.6	no	161.9	417.2
5	6.6	yes	108.4	445.6
	Average		141.4	449.2
	Median		143.5	431.1
	Std. Dev.		29.8	48.0



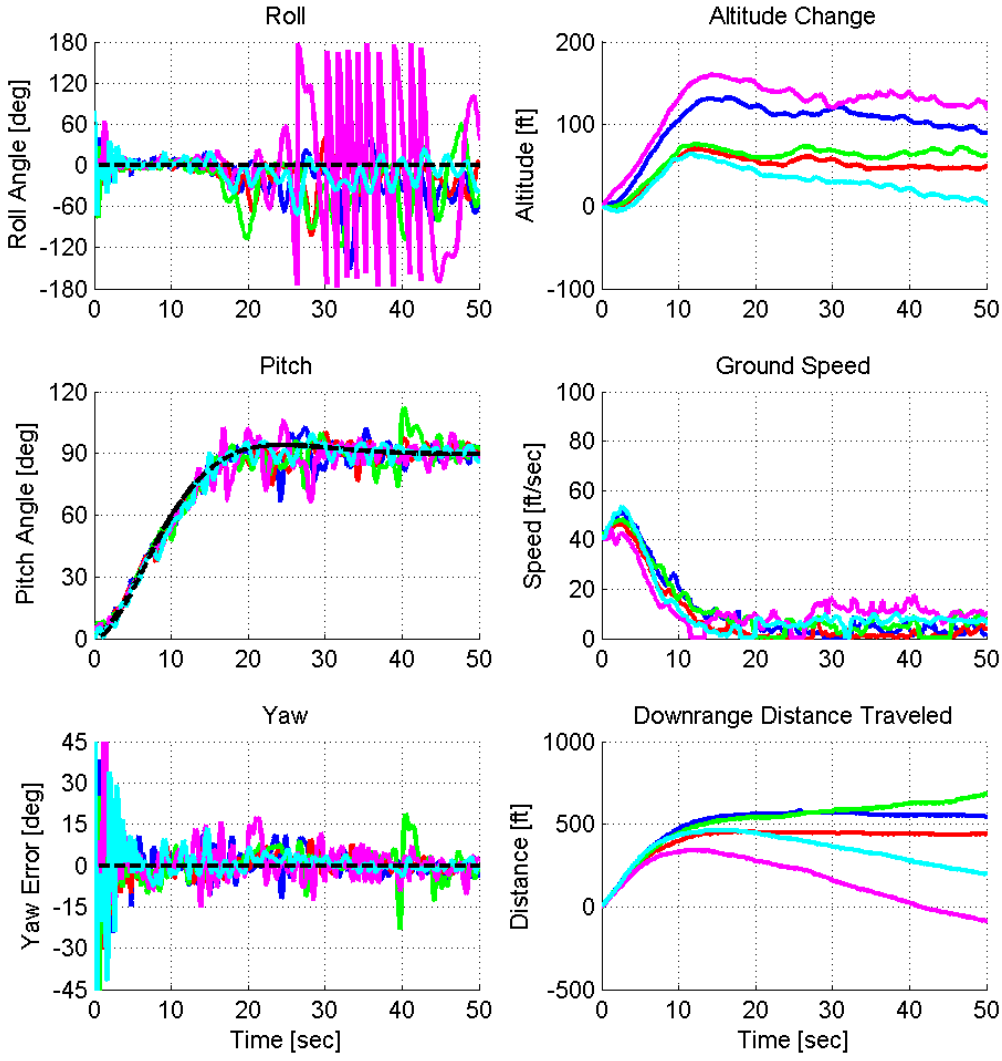
Flight Test: MRAC, Rise Time = 7 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.7	yes	170.1	531.7
2	0.0	yes	152.5	743.6
3	3.3	yes	191.3	584.9
4	1.0	no	91.4	567.6
5	0.0	yes	147.0	595.2
	Average		165.2	613.9
	Median		161.3	590.0
	Std. Dev.		20.0	90.9



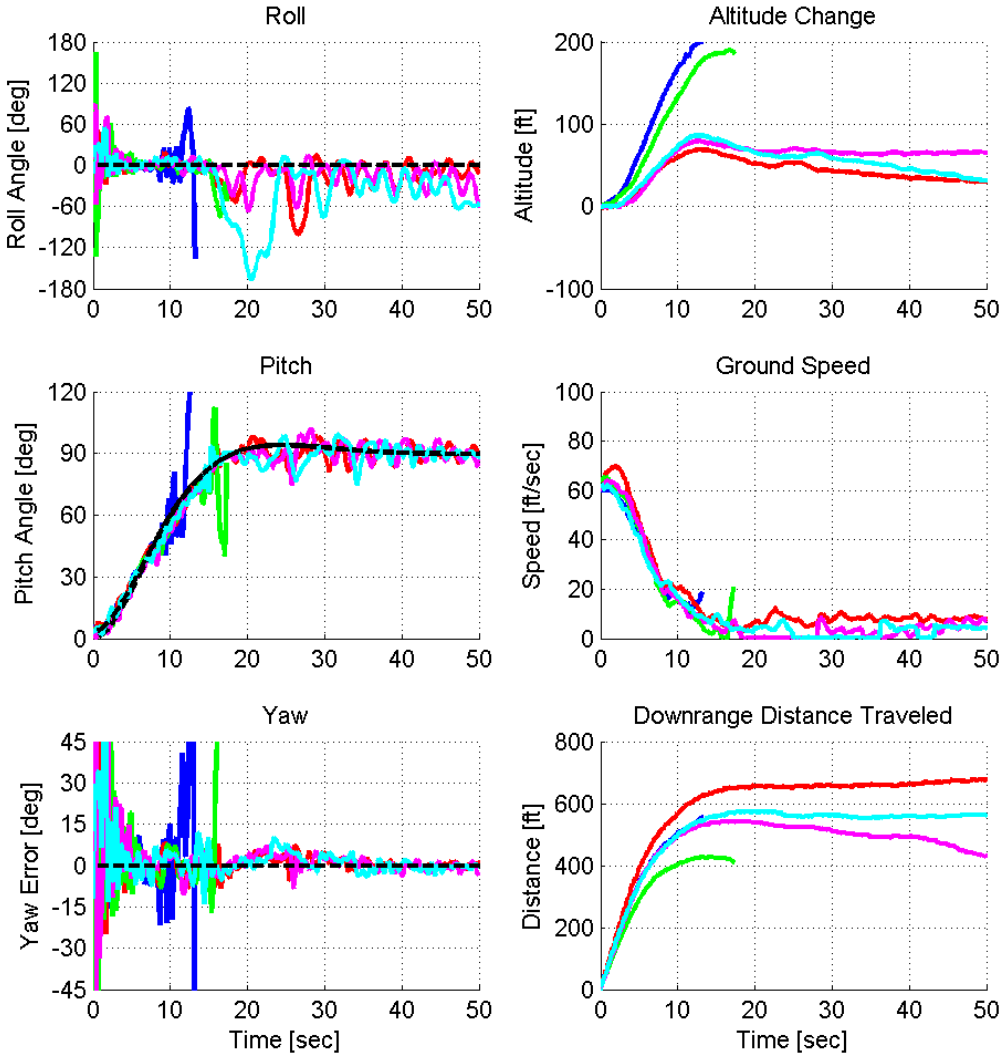
Flight Test: MRAC, Rise Time = 10 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	11.8	yes	132.1	578.1
2	14.4	yes	70.0	453.2
3	5.9	yes	75.7	682.0
4	9.5	yes	160.2	341.5
5	15.1	yes	63.8	462.8
	Average		100.4	503.5
	Median		75.7	462.8
	Std. Dev.		43.2	130.3



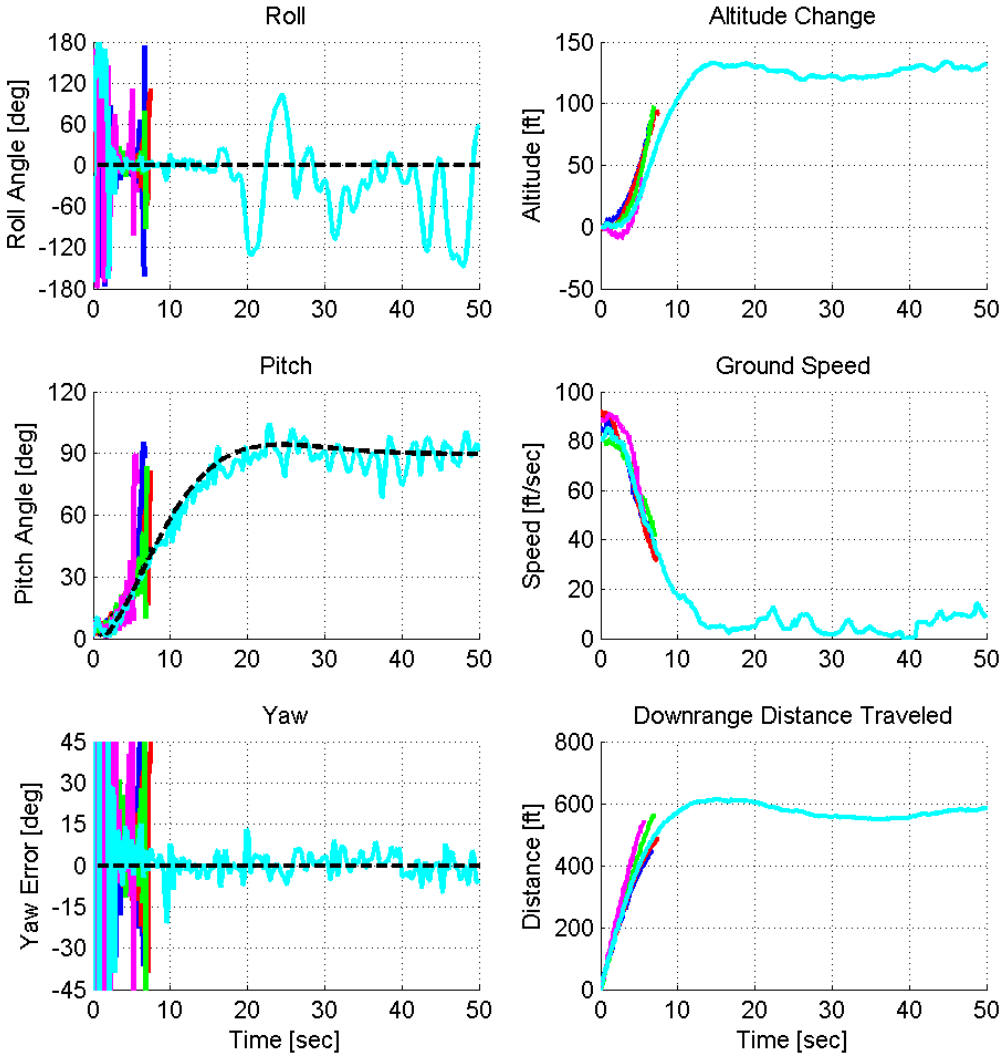
Flight Test: MRAC, Rise Time = 10 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	2.3	no	200.0	554.8
2	8.9	yes	69.1	677.7
3	2.6	no	190.0	426.7
4	12.1	yes	79.7	542.6
5	3.6	yes	86.3	576.6
	Average		78.4	599.0
	Median		79.7	576.6
	Std. Dev.		8.6	70.2



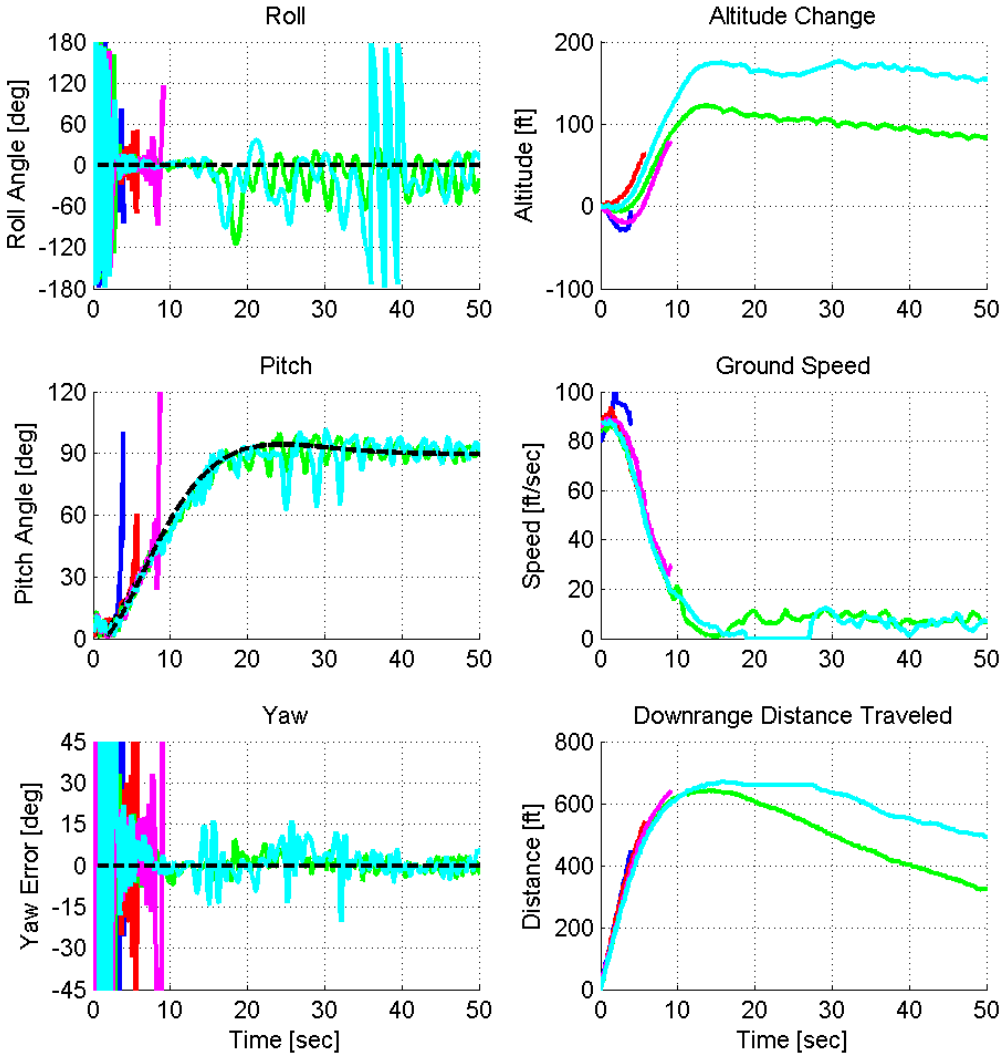
Flight Test: MRAC, Rise Time = 10 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	6.2	no	83.3	449.4
2	3.0	no	94.6	484.6
3	4.9	no	97.8	565.2
4	2.6	no	39.3	544.8
5	0.0	yes	133.7	612.8
	Average		133.7	612.8
	Median		133.7	612.8
	Std. Dev.		N/A	N/A



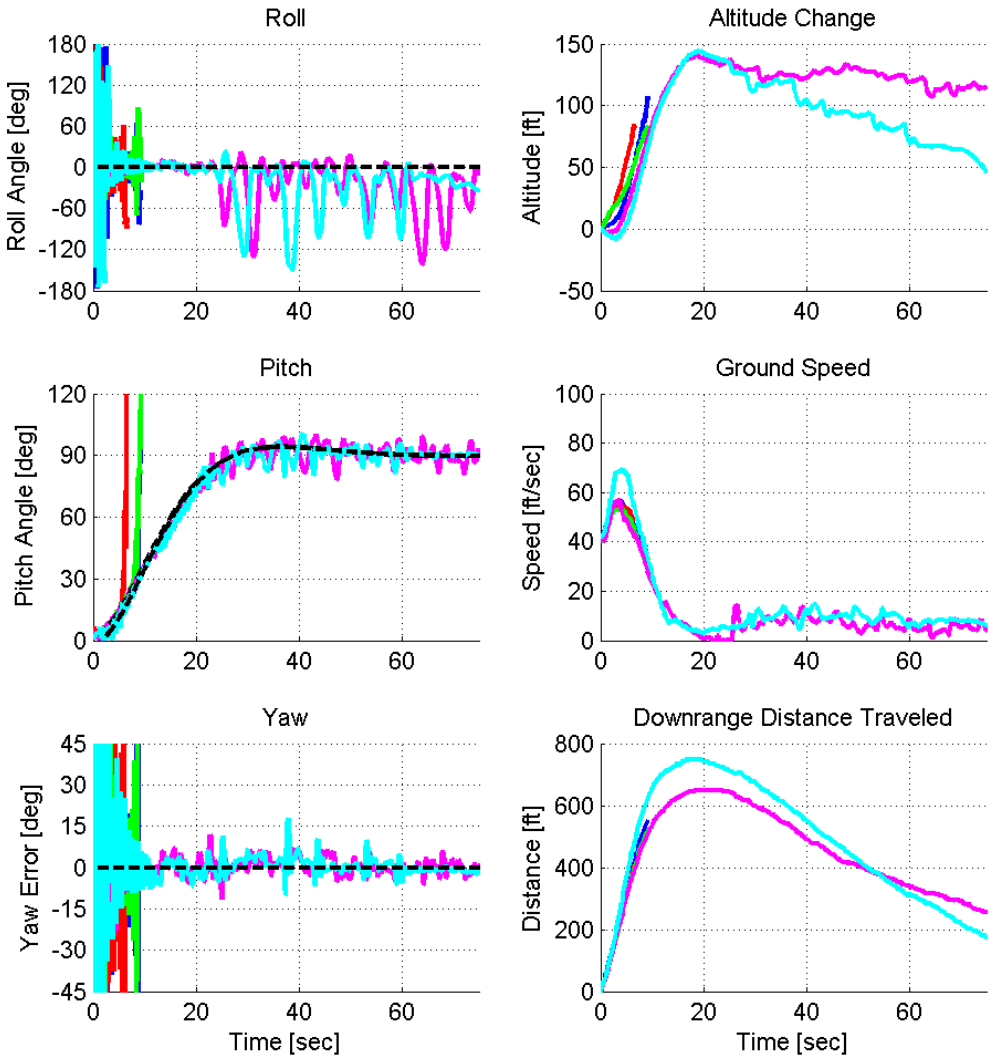
Flight Test: MRAC, Rise Time = 10 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	5.6	no	0.0	447.1
2	1.6	no	62.8	542.8
3	5.6	yes	122.2	640.6
4	2.3	no	77.3	640.2
5	0.0	yes	176.0	669.0
	Average		149.1	654.8
	Median		149.1	654.8
	Std. Dev.		38.0	20.0



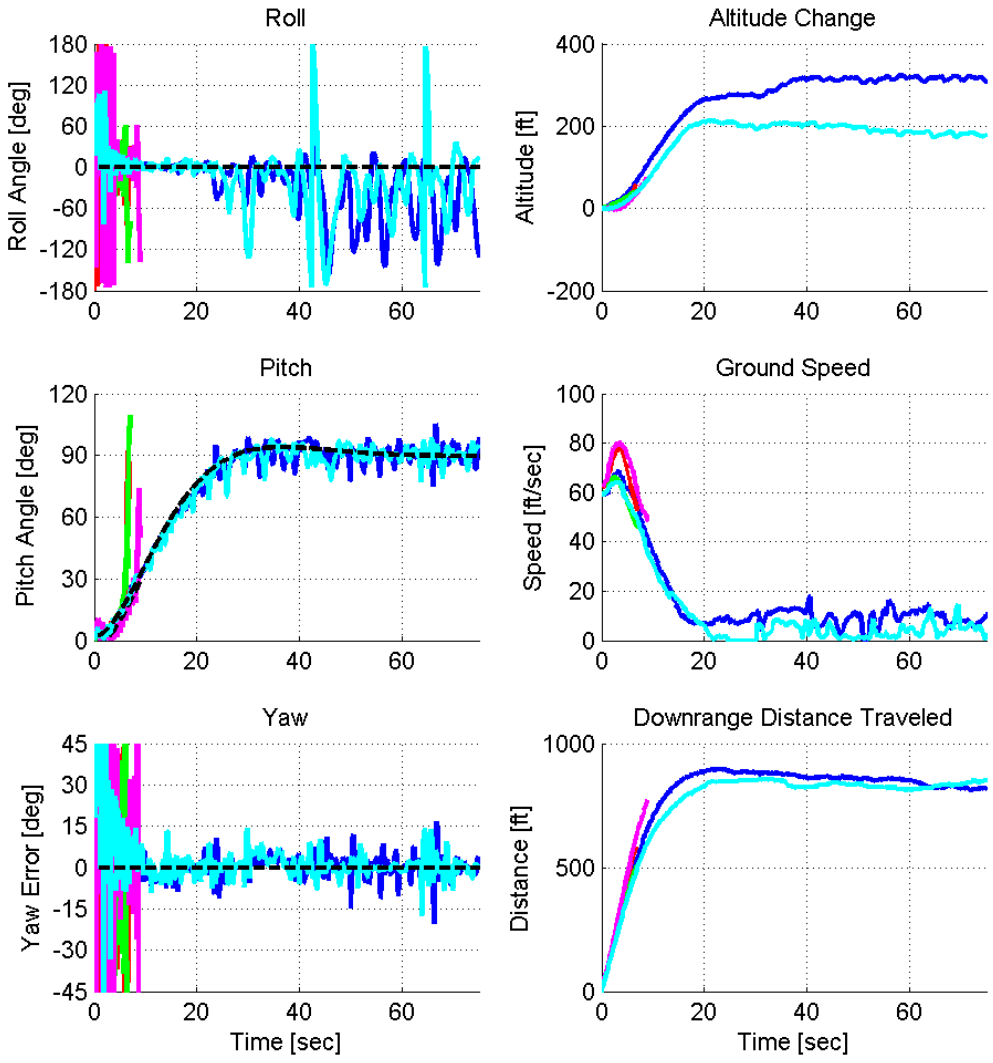
Flight Test: MRAC, Rise Time = 15 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	2.6	no	107.7	553.5
2	0.7	no	84.6	410.3
3	0.0	no	83.5	520.5
4	8.9	yes	141.2	649.0
5	8.9	yes	144.6	749.7
	Average		142.9	699.4
	Median		142.9	699.4
	Std. Dev.		2.4	71.2



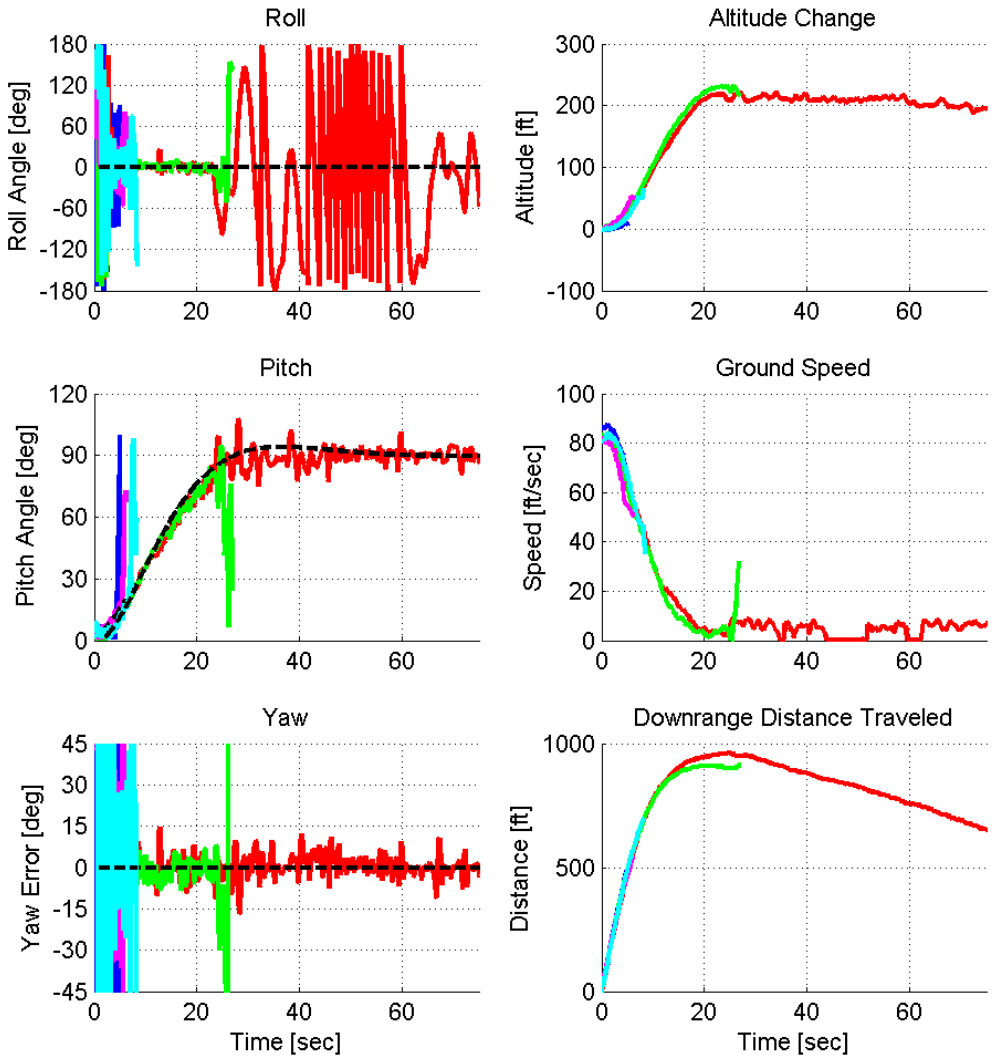
Flight Test: MRAC, Rise Time = 15 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	yes	324.4	897.4
2	0.7	no	57.5	582.3
3	6.2	no	41.8	508.9
4	3.6	no	74.1	770.4
5	6.9	yes	213.0	856.6
	Average		268.7	877.0
	Median		268.7	877.0
	Std. Dev.		78.8	28.9



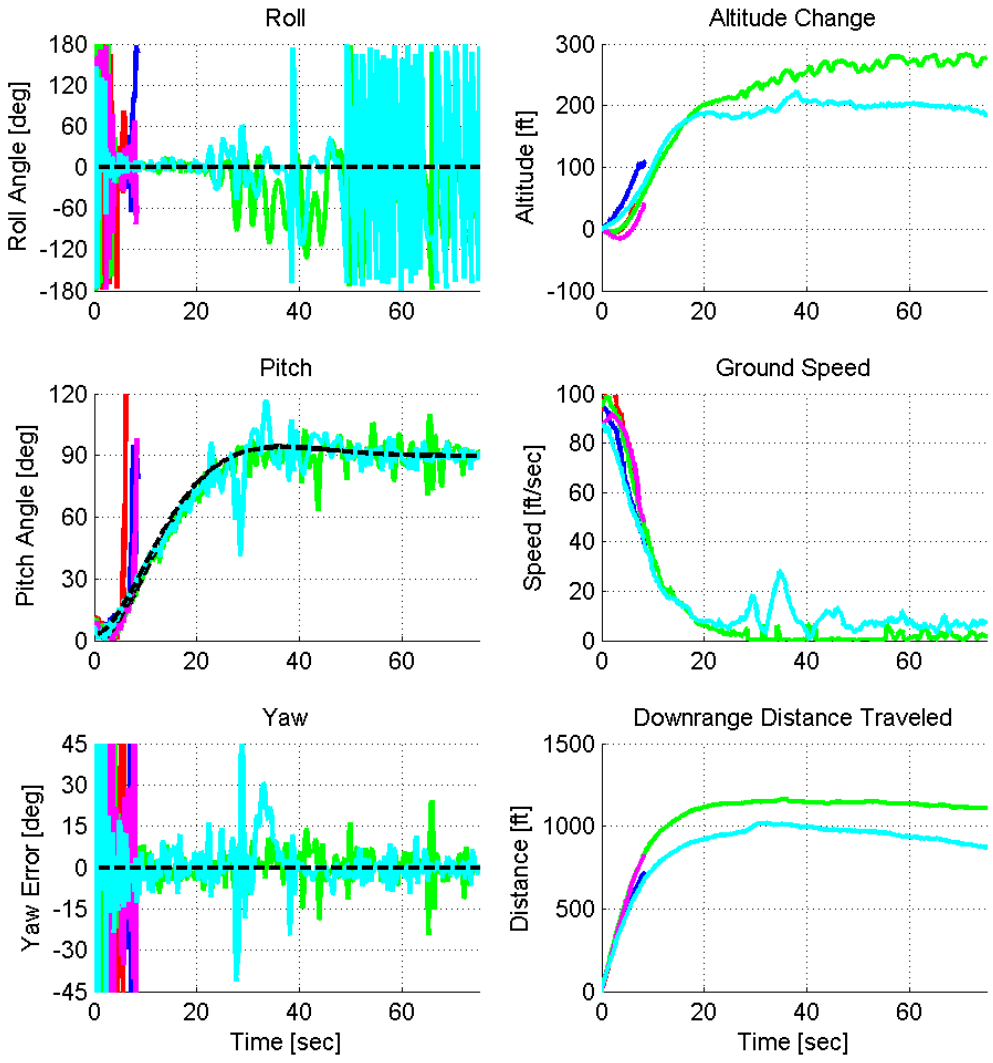
Flight Test: MRAC, Rise Time = 15 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	6.9	no	10.0	490.1
2	7.9	yes	221.3	960.1
3	9.2	no	232.2	920.0
4	3.6	no	53.8	542.8
5	6.2	no	66.0	700.1
	Average		221.3	960.1
	Median		221.3	960.1
	Std. Dev.		N/A	N/A



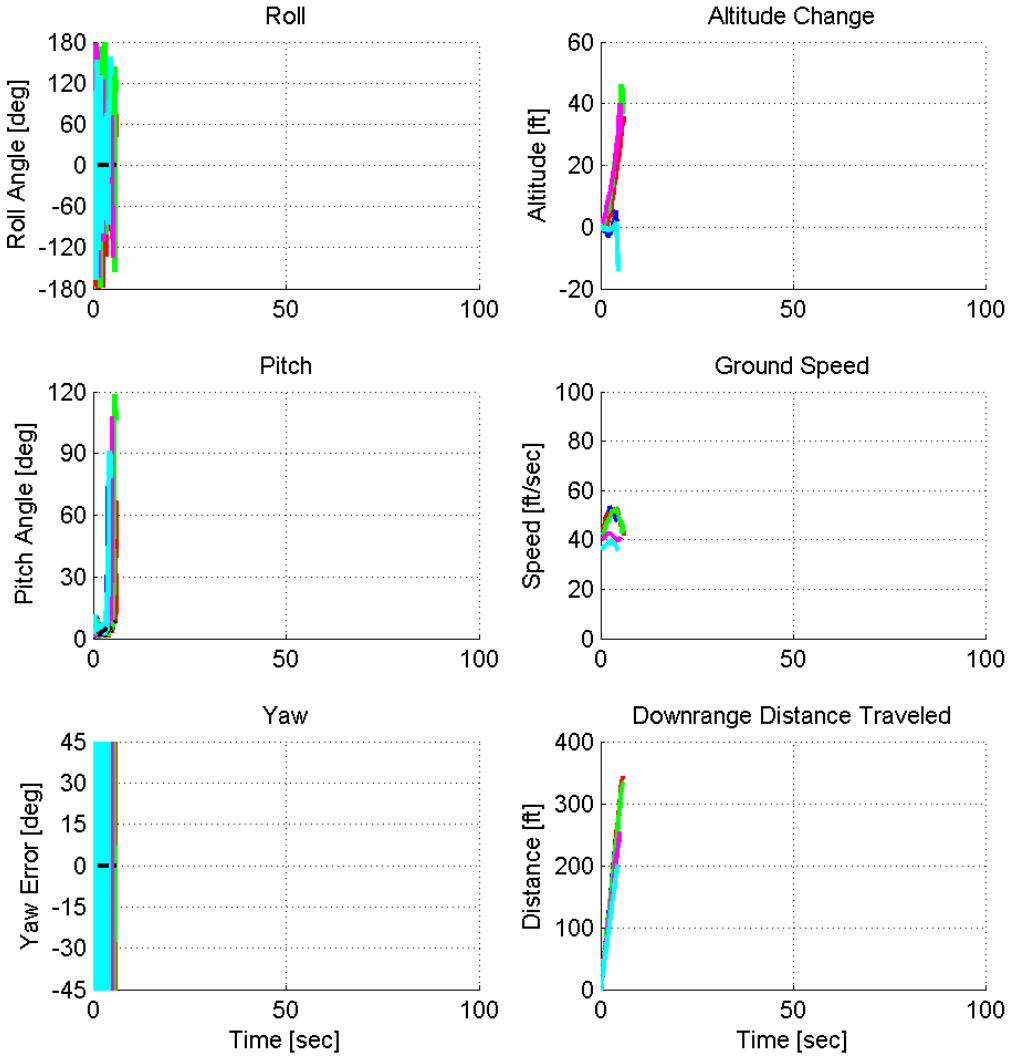
Flight Test: MRAC, Rise Time = 15 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	0.0	no	109.7	722.3
2	2.0	no	42.0	684.2
3	2.3	yes	282.7	1158.6
4	7.9	no	41.3	826.7
5	1.3	yes	223.0	1018.2
	Average		252.9	1088.4
	Median		252.9	1088.4
	Std. Dev.		42.2	99.2



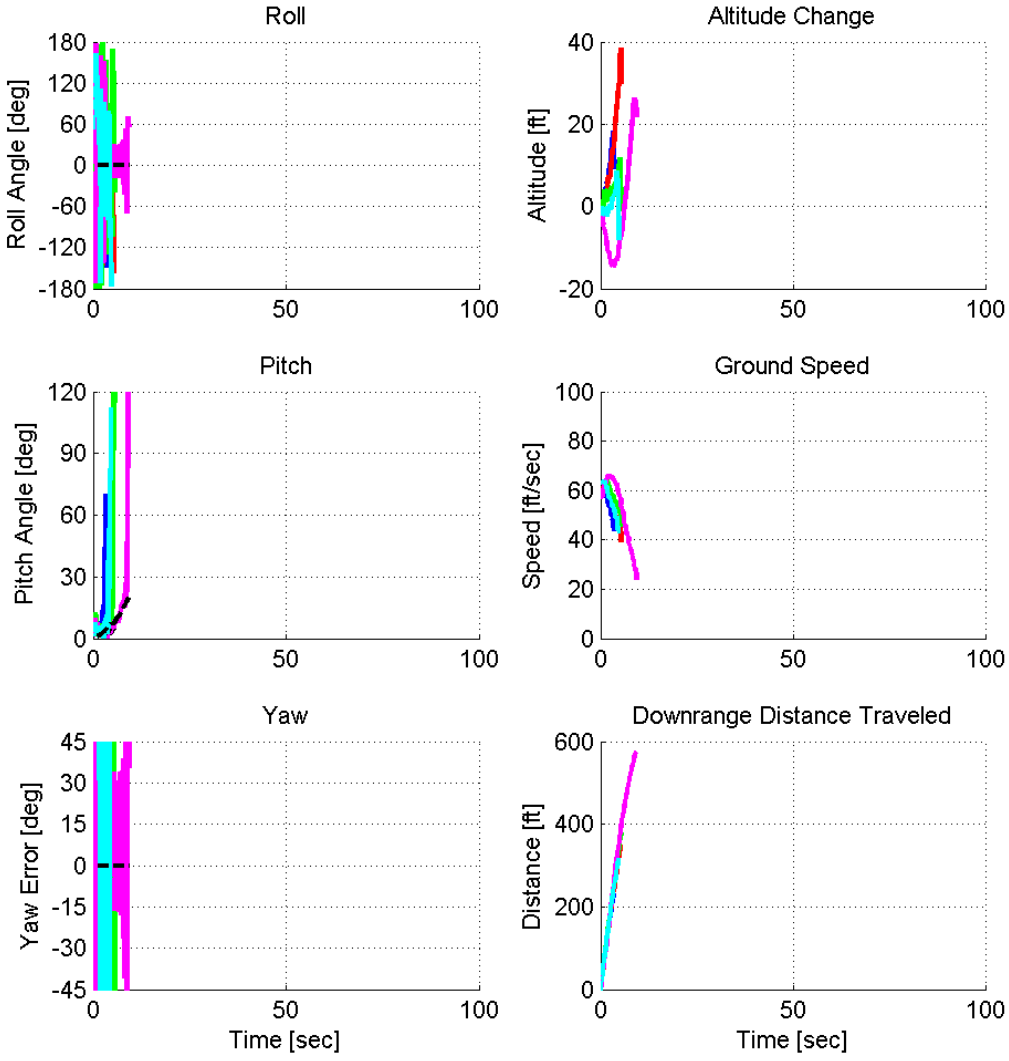
Flight Test: MRAC, Rise Time = 20 sec

Approach Speed: 40 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	7.2	no	5.4	251.2
2	7.2	no	35.9	343.1
3	5.6	no	46.2	332.8
4	9.8	no	40.3	255.2
5	1.0	no	1.6	201.2
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



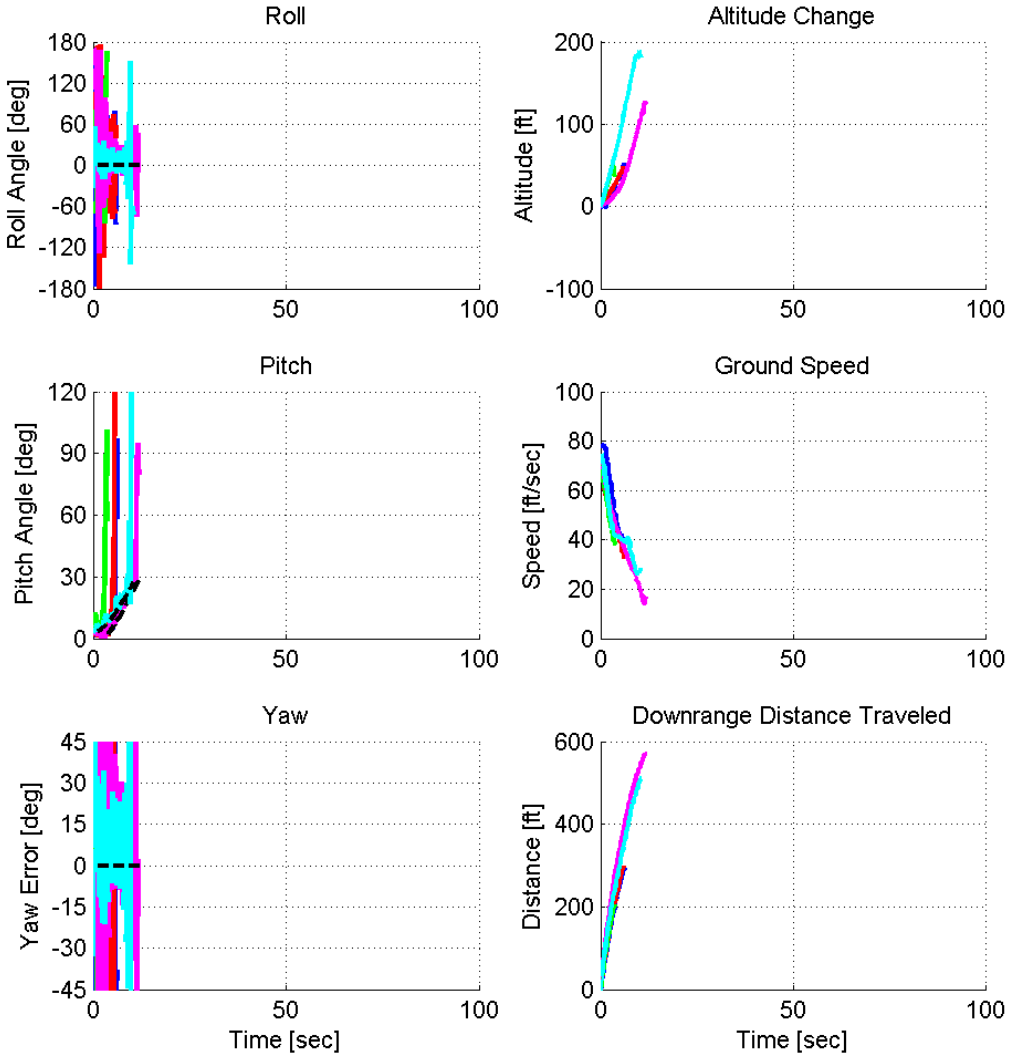
Flight Test: MRAC, Rise Time = 20 sec

Approach Speed: 60 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	3.6	no	18.3	230.7
2	8.5	no	38.5	350.2
3	0.0	no	12.0	379.2
4	0.3	no	26.2	573.4
5	7.9	no	8.6	318.7
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



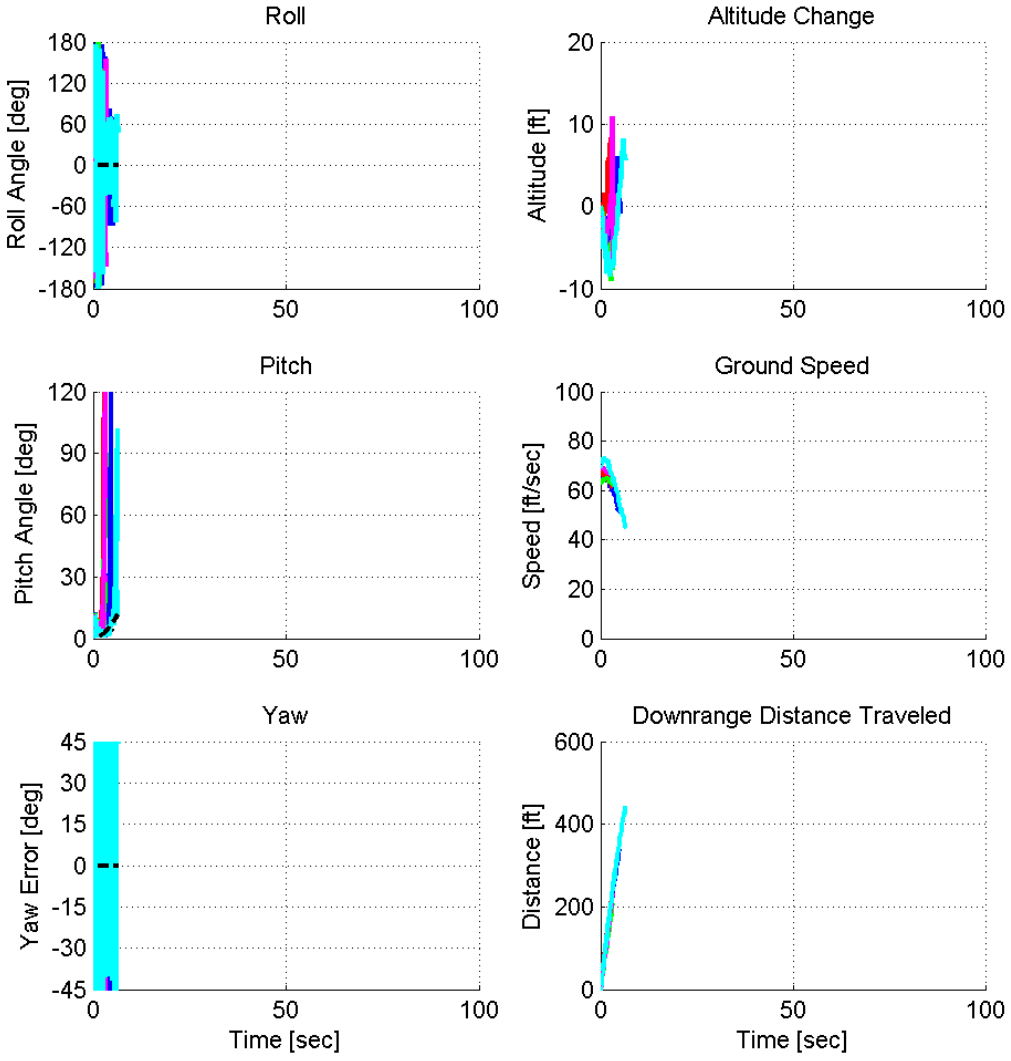
Flight Test: MRAC, Rise Time = 20 sec

Approach Speed: 80 ft/sec				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	4.3	no	52.5	294.5
2	14.8	no	48.4	298.3
3	9.5	no	48.8	205.8
4	3.6	no	127.8	569.2
5	6.2	no	189.1	509.4
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



Flight Test: MRAC, Rise Time = 20 sec

Approach Speed: ∼100 ft/sec (Max. Throttle)				
Test No.	Wind [ft/sec]	Successful?	Max. Alt. [ft]	Max. Dist. [ft]
1	8.5	no	6.0	339.7
2	0.0	no	8.3	221.8
3	4.6	no	0.0	208.2
4	1.0	no	11.0	231.7
5	1.6	no	8.2	442.3
	Average		N/A	N/A
	Median		N/A	N/A
	Std. Dev.		N/A	N/A



B.4 Hover-to-Level Transition

Hover-to-Level Flight			
Test No.	Wind [ft/sec]	Successful?	Max. Altitude Change [ft]
1	3.2	yes	-97.0
2	4.0	yes	-33.2
	Average		-65.1
	Median		-65.1
	Std. Dev.		45.1